Universidad Autónoma de Madrid

Escuela Politécnica Superior

Máster Universitario en Investigación e Innovación en Inteligencia Computacional y Sistemas Interactivos

Algoritmos de aprendizaje automático para la clasificación de datos funcionales

Autor:
Luis Sánchez Calvo

Tutores:
Alberto Suárez González
Departamento de Ingeniería Informática

José Luis Torrecilla Noguerales
Departamento de Matemáticas

2 de septiembre de 2020

Abstract

In this work we address the issue of functional data classification with the aim of showing which approach we have to use to model the problem (functional or multivariate) to obtain the best results. We begin showing the necessary theoretical foundation (functional analysis and stochastic process theory) to define functional data and understand the tools to manage it. Then, we use some data analysis and processing techniques that take advantage of their functional nature to generate classifiers in a battery of data sets. Finally, we show the results obtained by the trained classifiers using techniques that take into account functional properties and others that do not, to reach the conclusion that a hybrid approach is the best option.

Resumen

En este trabajo vamos a abordar el problema de la clasificación de datos funcionales con el objetivo de mostrar cuál es el enfoque que hemos de utilizar para modelizar el problema (funcional o multivariante) para obtener los mejores resultados. Comenzamos estudiando el fundamento teórico necesario (análisis funcional y teoría de procesos estocásticos) para definir los datos funcionales y entender las herramientas para manejarlos. Después, usamos algunas técnicas de análisis y procesamiento de datos que aprovechan su naturaleza funcional para generar clasificadores que evaluamos mediante una batería de conjuntos de datos. Por último, mostramos los resultados obtenidos por los clasificadores entrenados mediante técnicas que tienen en cuenta las propiedades funcionales y otras que no lo hacen, para llegar a la conclusión de que un enfoque híbrido es la mejor opción.

Índice general

1. Introducción **2**
- 1.1. Análisis en espacios funcionales . . . 3
  - 1.1.1. Espacios de clases de funciones L^p . . . 3
  - 1.1.2. Serie de Fourier . . . 4
- 1.2. Procesos estocásticos . . . 5
- 1.3. Diferencias entre la estadística multivariante y la funcional . . . 7

2. Clasificación supervisada **9**
- 2.1. Aprendizaje supervisado: Clasificación . . . 9

3. Metodologías **14**
- 3.1. Random Forest . . . 14
- 3.2. Máquinas de Vector Soporte . . . 16
- 3.3. Procesamiento de datos . . . 17

4. Experimentos y resultados **20**
- 4.1. Brownianos con distinta esperanza . . . 21
- 4.2. Berkeley . . . 38
- 4.3. Phoneme . . . 55
- 4.4. Conclusiones . . . 71

5. Apéndices **72**
- 5.1. Demostración del Teorema 2.4 . . . 72
- 5.2. Gráficas adicionales . . . 74
  - 5.2.1. Experimento Brownianos con distinta esperanza . . . 74
  - 5.2.2. Experimento Berkeley . . . 76
  - 5.2.3. Experimento Phoneme . . . 78

Capítulo 1

Introducción

Gracias al aumento de capacidad de cálculo y de memoria de los ordenadores actuales, podemos almacenar y analizar conjuntos de datos que hace varios años sería imposible plantearse. Además, cada vez en más áreas de la ciencia, los datos con los que se trabaja son de naturaleza funcional.

> There is actually an increasing number of situations coming from different fields of applied sciences (environmetrics, chemometrics, biometrics, medicine, econometrics, ...) in which the collected data are curves. Indeed, the progress of the computing tools, both in terms of memory and computational capacities, allows us to deal with large sets of data.
>
> Non-parametric Functional Data Analysis. Theory and Practice - F. Ferraty & P. Vieu - Página 5 - [22]

Sin embargo, en ocasiones los datos funcionales se manejan y analizan utilizando herramientas, algoritmos o métodos que no aprovechan, o no tienen en cuenta, su naturaleza funcional. Un claro ejemplo de esto se puede ver en el artículo [29]. Aquí, los autores proponen procesar una serie de electrocardiogramas utilizando métodos multivariantes que no explotan las propiedades funcionales de los datos, para su posterior clasificación. Otro ejemplo puede encontrarse en el artículo [24], en el cual se se presenta un método de selección de variables específicamente diseñado para resolver problemas de clasificación.

No obstante, también se han diseñado procedimientos que sí que aprovechan algunas propiedades funcionales de los datos para mejorar la clasificación. En el artículo [3] se muestra cómo se pueden aplicar los *Reproducing Kernel Hilbert Spaces* (RKHS) en problemas de clasificación con datos funcionales. Por otro lado, en los artículo [20] y [28] se muestran diferentes maneras de modificar el algoritmo *Random Forest* para que tenga en cuenta algunas propiedades de los datos funcionales a la hora de entrenar un clasificador.

Nuestro objetivo en este trabajo es mostrar si se aprecian diferencias a la hora de modelizar un problema de clasificación con datos funcionales utilizando herramientas que tengan en cuenta estas propiedades frente a otras que no las usan. Para lograr este objetivo vamos a modelizar varios problemas funcionales usando los dos enfoques, y veremos cómo aquellos en los que se aplica el enfoque híbrido (en parte funcional y en parte multivariante) obtienen los mejores resultados.

Este trabajo está dividido en dos partes. La primera está dedicada a estudiar la teoría que sustenta el análisis de datos funcionales, mientras que la segunda se centra en la aplicación de técnicas funcionales de procesamiento y análisis, con el objetivo de resolver una batería de problemas de clasificación supervisada con datos funcionales. Comenzaremos por definir, en la Sección 1.1, los espacios en los que asumiremos que viven nuestros datos, y sus propiedades (Sección 1.1.1). Después, en la Sección 1.2, introduciremos la teoría de los procesos estocásticos, y terminaremos mostrando (en la Sección 1.3) las diferencias más significativas entre la estadística multivariante y la estadística con datos funcionales. Ya en la segunda parte del trabajo, en el Capítulo 2, se introduce el problema de clasificación supervisada. En el 3 se describen los algoritmos de aprendizaje supervisado junto con algunas técnicas vistas en la literatura utilizadas para procesar y analizar los datos funcionales. Esto lo usaremos en el Capítulo 4 afrontar un conjunto de problemas de clasificación supervisada (experimentos) utilizando los dos enfoques. Por último analizaremos los resultados de los experimentos y veremos como los mejores resultados son obtenidos a partir de un enfoque híbrido.

1.1. Análisis en espacios funcionales

Si quisiéramos analizar el comportamiento del crecimiento de las personas en una población a lo largo del tiempo podríamos modelizar el problema aplicando dos enfoques diferentes. El primero consiste en asumir que los datos que se recogen son funciones que dependen de una variable temporal t (por ello llamaremos a este enfoque *funcional*). De esta manera trabajaremos con un conjunto de datos funcionales $\{x_j = x_j(t); \quad j = 0, \ldots, n-1\}$, donde n denota el tamaño de la muestra y x_j representa la altura de la persona j-ésima a lo largo del tiempo. El segundo enfoque (al que llamaremos *multivariante*) consiste en tomar una discretización de la variable temporal t y trabajar con los vectores de mediciones $\{x_{j,i} = x_j(t_i); \quad j = 0, \ldots, n-1.i = 0, \ldots, d-1\}$. Ahora el vector x_j contiene las mediciones de las alturas de la persona j en los tiempos t_i siendo d es el número de mediciones. La principal diferencia entre los dos enfoque radica en que, en el funcional, el objeto de estudio las funciones $x_j(t)$ mientras que en el multivariante son los vectores $x_j(t_i)$.

Más adelante definiremos un dato funcional como una realización de un proceso estocástico que toma valores en un espacio de funciones, pero para poder entender lo que esto significa antes necesitamos hablar de la teoría del análisis en espacios funcionales. Para ello necesitamos conocer, entre otras cosas, qué son estos espacios de funciones en los que vamos a trabajar, y sus propiedades (Sección 1.1.1). Además, en la Sección 1.1.2, introduciremos el concepto de serie de Fourier de una función, herramienta que utilizaremos más adelante para representar los datos funcionales en la base de Fourier.

1.1.1. Espacios de clases de funciones L^p

En esta sección estudiaremos los espacios de clases de funciones (o simplemente espacios de funciones) L^p para motivar la importancia del espacio L^2. Este espacio tiene unas propiedades muy interesantes y por eso asumiremos que es el espacio en el que viven nuestros datos funcionales cuando nos enfrentemos a un problema de clasificación funcional.

Cuando analicemos un problema de clasificación supervisada funcional los elementos que vamos a querer estudiar van a ser funciones, y para poder entender mejor su comportamiento y las herramientas con las que vamos a tratarlas, vamos a definir el espacio en el que asumiremos que viven. Para ello definimos el concepto de *espacio de funciones* o *espacio funcional*.

Definición 1.1. [31, c. 3 p. 65] *Dado $1 \leq p < \infty$ y un conjunto X, definimos el espacio de funciones $\mathcal{L}^P(X)$ como*

$$\mathcal{L}^p(X) = \left\{ f : X \longrightarrow \mathbb{C} \text{ medibles con } \int_X |f|^p < \infty \right\}.$$

Además definimos el espacio $L^P(X)$ como $\mathcal{L}^p(X)$ con la relación de equivalencia

$$f \sim g \Longleftrightarrow f = g \text{ en casi todo punto, es decir si } \int_X f - g = 0.$$

Por como hemos definido estos espacios de funciones podemos ver que L^p es un espacio vectorial, ya que:

1. $\forall f, g \in L^p$ se tiene que $f + g \in L^p$.
2. $\forall f \in L^p$ y $\forall \lambda \in \mathbb{C}$ se tiene que $\lambda f \in L^p$.

Además, si en estos espacios consideramos las normas

$$\|f\|_p = \left(\int_X |f|^p \right)^{1/p}, \tag{1.1}$$

entonces se obtiene que $(L^p(X), \|\cdot\|)$ es un espacio normado, ya que cumple las propiedades:

1. $\|f\|_p \geq 0$ y $\|f\|_p = 0 \Longleftrightarrow f = 0$ en c.t.p.
2. $\|\lambda f\|_p = |\lambda| \|f\|_p$.
3. $\|f + g\|_p \leq \|f\|_p + \|g\|_p$[1].

[1] Esta última propiedad (triangular) se tiene gracias a la desigualdad de Hölder.

Por lo que acabamos de ver, los espacios de funciones L^p tienen una estructura bastante rica: son espacios vectoriales en los cuales podemos definir una norma. De entre todos espacios L^p, hay uno que destaca, y es el caso de el espacio de funciones L^2. Este espacio destaca frente al resto de espacios L^p ya que además podemos definir un producto interno de la siguiente forma.

Definición 1.2. [31, c. 4 p. 78] *En el espacio $L^2(X)$ definimos el producto interno (o producto escalar) de dos funciones f y g como*

$$\langle f, g\rangle = \int_X f(t)\overline{g}(t)dt,$$

donde $\overline{g}(t)$ denota el conjugado complejo de la función $g(t)$.

Gracias a esta definición del producto escalar, podemos ver que se cumplen las siguientes propiedades:

1. $\langle f, f\rangle \geq 0$ y $\langle f, f\rangle = 0 \iff f = 0$ en c.t.p.
2. $\overline{\langle f, g\rangle} = \langle g, f\rangle$.
3. $\forall \lambda \in \mathbb{C}, \langle \lambda f, g\rangle = \lambda\langle f, g\rangle$.
4. $\langle f_1 + f_2, g\rangle = \langle f_1, g\rangle + \langle f_2, g\rangle$.

Gracias a que el producto interno cumple estas cuatro propiedades se tiene que el espacio L^2 es un espacio *pre-Hilbert*. Si a esto le añadimos que L^2 es completo (esto es, que toda sucesión de Cauchy tiene límite dentro del propio espacio L^2) entonces decimos que L^2 es un espacio de *Hilbert*.

1.1.2. Serie de Fourier

En ocasiones querremos representar nuestros datos en una base de funciones. Una de las bases más populares es la base de Fourier. Para ver cómo podemos representar una función de $L^2([0,1))$ en esta base introducimos el concepto de serie de Fourier. Esta herramienta nos va a ser útil por que nos va a permitir codificar una función mediante un conjunto de coeficientes. En la Sección 3.3, aplicaremos esta técnica como método de reducción de dimensión.

La serie de Fourier de una función es la representación de esta en una base especial, pero antes de definir en qué consiste esta serie, vamos a ver el siguiente teorema.

Teorema 1.3. [31, c. 4 p. 89] *En el espacio $L^2([0,1)) = \{f : [0,1) \longrightarrow \mathbb{C},$ medibles con $\int_0^1 |f|^2 dx < \infty\}$, se tiene que $\{e^{2\pi inx}\}_{n\in\mathbb{Z}}$ es un sistema ortonormal completo.*

Gracias a este teorema tenemos que toda función de $L^2([0,1))$ se puede escribir en la base $\{e^{2\pi inx}\}_{n\in\mathbb{Z}}$ (también llamada base de Fourier). La serie de Fourier de una función no es más que su representación en esta base.

Definición 1.4. [31, c. 5 p. 103] *La serie de Fourier de una función f de $L^2([0,1))$ se define como*

$$\sum_{n\in\mathbb{Z}} \widehat{f}(n)e^{-2\pi inx},$$

donde

$$\widehat{f}(n) = \langle f, e^{2\pi inx}\rangle = \int_0^1 f(x)e^{-2\pi inx}dx.$$

Ejemplo 1. [16] *Sea $f(x) = x$ en el intervalo $[0,1)$ extendida periódicamente. Como f pertenece a $L^2([0,1))$, ya que $\int_0^1 x^2dx = 1/3 < \infty$, entonces f admite una representación en la base de Fourier. Los coeficientes de la serie de Fourier de esta función son*

$$\widehat{f}(0) = \langle f, 1\rangle = \int_0^1 xdx = \frac{1}{2}, \text{ si } n = 0. \tag{1.2}$$

Por otro lado, si $n \neq 0$, integrando por partes se llega a que

$$\widehat{f}(n) = \langle f, e^{2\pi inx}\rangle = \int_0^1 xe^{-2\pi inx}dx = \left[\frac{x}{-2\pi in}e^{2\pi inx}\right]_0^1 + \frac{1}{2\pi inx}\int_0^1 e^{2\pi inx}dx = \frac{i}{2\pi n}. \tag{1.3}$$

Por el Teorema 1.3, se concluye que f se puede representar de la siguiente forma en su serie de Fourier:

$$f(x) = \frac{1}{2} + \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} \frac{i}{2\pi n} e^{2\pi i n x}. \tag{1.4}$$

Gracias al Teorema 1.3, vemos cómo podemos escribir una función en forma de serie tan solo conociendo sus coeficientes en la base de Fourier. Recíprocamente, si conocemos los coeficientes en la base de Fourier, podemos reconstruir la función original. Esta identificación entre la función y el conjunto de coeficientes de Fourier será el pilar fundamental de una de las técnicas de reducción de dimensión que utilizaremos más adelante.

1.2. Procesos estocásticos

Al modelizar un problema de clasificación con un enfoque funcional asumiremos que los datos funcionales son observaciones de un proceso estocástico que toma valores en un espacio de funciones (de la misma forma que en la clasificación multivariante suponíamos que cada vector de la muestra se correspondía con una observación de una variable aleatoria que toma valores en $\mathbb{R}^d$). Por ello la teoría de los procesos estocásticos es el fundamento teórico necesario para poder entender la procedencia de nuestros datos.

Definición 1.5. [2, c. 6, p. 224] *Sea $\mathcal{T}$ un conjunto. Decimos que una familia de variables aleatorias $\{X(t), t \in \mathcal{T}\}$ es un proceso estocástico. Si $\mathcal{T}$ es finito o numerable entonces llamaremos a $\{X(t), t \in \mathcal{T}\}$ proceso estocástico en tiempo discreto. Si $\mathcal{T}$ es un intervalo entonces diremos que $\{X(t), t \in \mathcal{T}\}$ es un proceso estocástico a tiempo continuo.*

Uno de los procesos estocásticos más populares es el conocido como *proceso browniano*, *movimiento browniano* o *proceso de Wiener*. Antes de mostrar en que consisten este tipo de procesos, necesitamos definir el concepto de *proceso gaussiano*.

Definición 1.6. [17, c. 3, p. 59] *Decimos que un proceso estocástico $\{X_t\}_{t\in\mathcal{T}}$, con $\mathcal{T}$ un intervalo, es gaussiano si cualquier vector $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ tiene una distribución gaussiana multivariante. Es decir, cualquier proyección finito dimensional del proceso se distribuye como una normal multivariante.*

Con esta última definición ya tenemos las herramientas necesarias para definir lo que es un proceso browniano estándar.

Definición 1.7. [17, c. 5, p. 95] *Un proceso estocástico $\{X_t\}_{0\leq t\leq T}$ se dice que es un movimiento browniano estándar si*

1. *X_t es un proceso gaussiano.*

2. *$\mathbb{E}[X_t] = 0$ y $\mathbb{E}[X_t X_s] = \min(t, s)$.*

3. *Para casi todo suceso ω, el camino $t \mapsto X_t(\omega)$ es continuo en $[0, T]$.*

En la Figura 1.1 pueden verse dos observaciones de procesos brownianos, mientras que en la Figura 1.2 se muestran las medias y desviaciones típicas de un conjunto de observaciones junto con algunas trayectorias.

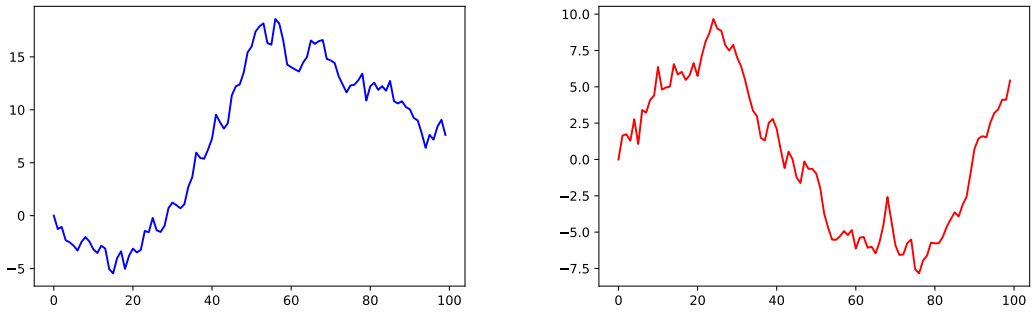

Figura 1.1: A la izquierda (azul) una observación de un proceso browniano estándar. A la derecha (rojo) una observación de un proceso browniano con $\mathbb{E}[X_t] = 10\sin(2\pi t/100)$.

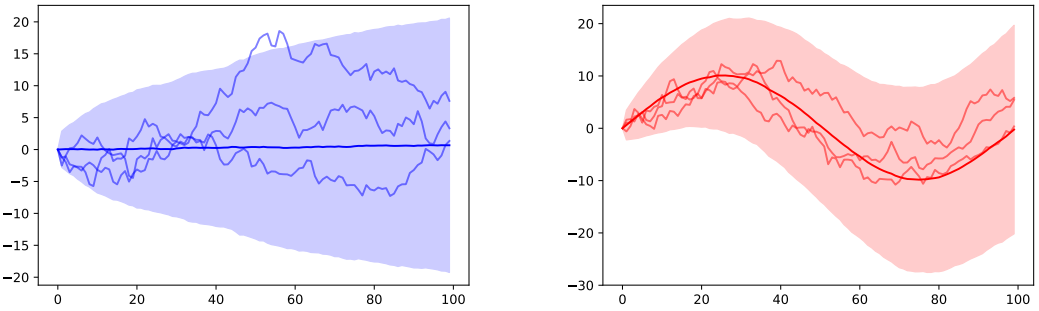

Figura 1.2: A la izquierda (azul) la media de 500 observaciones de procesos brownianos estándar $\pm$ 1 desviación típica y tres trayectorias. A la derecha (rojo) la media de 500 observaciones de procesos brownianos, con $\mathbb{E}[X_t] = 10\sin(2\pi t/100)$, $\pm$ una desviación típica y tres trayectorias.

Un proceso gaussiano queda definido completamente por su esperanza $\mathbb{E}[X_t]$ y función de covarianzas $\mathbb{E}[X_t, X_{t'}]$.

Este tipo de procesos estocásticos son muy populares y aparecen en una gran variedad de ramas de la ciencia (economía, química, física, biología...). Además, los utilizaremos en la segunda parte del trabajo para mostrar las diferencias entre modelizar un problema usando el enfoque funcional y el multivariante.

Cuando estudiemos un problema de clasificación funcional los datos que tendremos serán curvas (aunque por las limitaciones de almacenamiento de los ordenadores solo dispondremos de una cantidad finita de valores de estas trayectorias). Estas curvas (o trayectorias) las veremos como realizaciones de un proceso estocástico que toma valores en un espacio de funciones (generalmente en $L^2([0,1))$). En la Figura 1.3 se puede ver un ejemplo de dato funcional. Esta trayectoria corresponde a las mediciones de la temperatura en Madrid (El Retiro) medido todos los días del año 1997 a las 12 del mediodía.

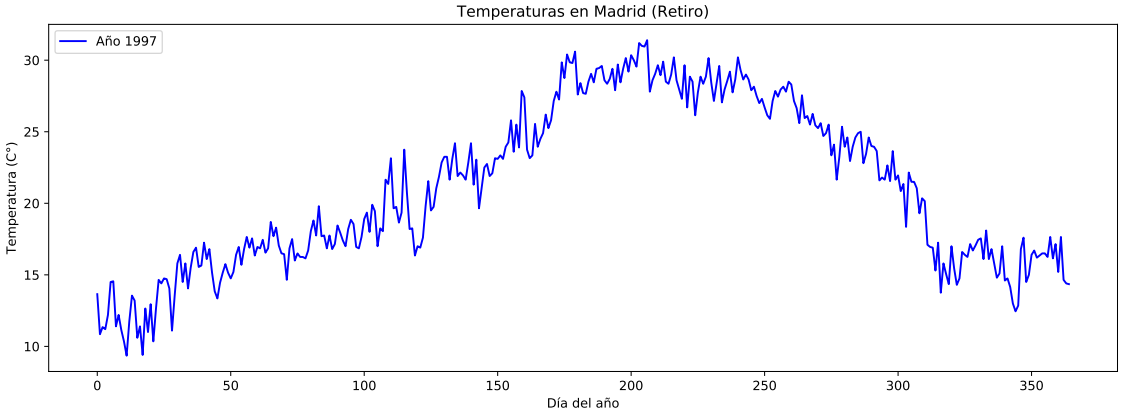

Figura 1.3: Temperatura del año 1997 en El Retiro (Madrid). Fuente: AEMET

1.3. Diferencias entre la estadística multivariante y la funcional

Al pasar de trabajar con variables aleatorias en un espacio vectorial de dimensión finita (enfoque multivariante) a hacerlo en uno de dimensión infinita (enfoque funcional), hay propiedades diferentes a tener en cuenta y conceptos que no se extienden de manera obvia. Una de las diferencias más notables tiene que ver con la continuidad. Mientras que un dato multivariante consta de una cantidad finita de atributos, los datos funcionales son trayectorias continuas. Además, el hecho de que no podamos definir el concepto de *función de densidad* cuando tratamos con un proceso estocástico trae consigo consecuencias importantes, como que la *regla Bayes* de un problema de clasificación funcional es mucho más compleja de definir. Otra consecuencia de este hecho involucra al concepto de la *moda*. La moda en el caso multivariante, se define como el elemento que tiene "mayor probabilidad" de ocurrir, es decir, el elemento que maximiza la función de densidad de probabilidad de la variable aleatoria en cuestión. Ahora bien, como en el caso de los procesos estocásticos no tenemos una noción de densidad, no podemos generalizar este concepto de manera directa. Se puede definir la moda de un proceso estocástico de diferentes maneras, sin embargo, una de las más sencillas es la siguiente:

Definición 1.8. [15] *Se define la $h-$moda, que llamaremos M_0, de un proceso estocástico $\mathcal{X}$ como*

$$M_0 = \arg\max_a \mathbb{E}K\left(\frac{\|a-\mathcal{X}\|}{h}\right),$$

donde h es una constante fijada de antemano, y K es una función kernel (núcleo).

Un concepto que puede extenderse al caso funcional de varias maneras es la esperanza de un proceso estocástico. Una de ellas es la siguiente

Definición 1.9. [7, c. 1, p. 27] *Sea $(\Omega, \mathcal{A}, \mu)$ un espacio de probabilidad y $\mathcal{B}$ un espacio de Banach*[2] *separable. Si una variable aleatoria $\mathcal{X}: \Omega \longrightarrow \mathcal{B}$ es Bochner-integrable, es decir, existe una sucesión de variable aleatorias simples $\mathcal{X}_n$ tal que*

$$\lim_{n\to\infty} \int_\Omega \|\mathcal{X} - \mathcal{X}_n\|_{\mathcal{B}} d\mu = 0, \tag{1.5}$$

entonces se define su esperanza de Bochner (o esperanza fuerte) como

$$\int_\Omega \mathcal{X} d\mu = \lim_{n\to\infty} \int_\Omega \mathcal{X}_n d\mu. \tag{1.6}$$

Sin embargo, también se puede definir la esperanza de un proceso aleatorio de la siguiente manera.

[2] Esto es un espacio vectorial normado y completo.

Definición 1.10. [7, c. 1, p. 28] *Sea $\mathcal{X}$ un proceso estocástico con valores en un espacio de Banach separable $\mathcal{B}$. Se dice que $\mathcal{X}$ es Pettis-integrable (o integrable débil) si existe un elemento de $\mathcal{B}$, al que llamaremos $E\mathcal{X}$, tal que $\mathbb{E}(b^*(\mathcal{X})) = b^*(E\mathcal{X})$ para todo elemento b^* del espacio dual de $\mathcal{B}$. En este caso, llamamos esperanza débil de $\mathcal{X}$ a $E\mathcal{X}$.*

Cada una de estas definiciones de la esperanza de un proceso estocástico aporta información diferente, y, dependiendo de las propiedades asintóticas que se quieran estudiar conviene utilizar una u otra. Por ejemplo, si tenemos una colección de n datos funcionales de L^2 $\{x_i(t)\}_{i=1}^n$ (esto es, n observaciones de un proceso estocástico $\mathcal{X}$ que toma valores en L^2) definimos el estimador de la esperanza de la forma

$$\widehat{m}_n(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(t), \tag{1.7}$$

es decir, mediante la media muestral. Si queremos obtener que

$$\lim_{n\to\infty} \|\widehat{m}_n - m\| = 0 \quad \text{casi seguro}^3, \tag{1.8}$$

necesitamos tomar $m = \mathbb{E}(\mathcal{X})$ la esperanza fuerte (Bochner). Sin embargo, si nos conformamos con

$$\lim_{n\to\infty} \langle \widehat{m}_n, b\rangle = \langle m, b\rangle \quad \forall b \in L^2, \tag{1.9}$$

es suficiente tomar $m = E\mathcal{X}$ la esperanza débil (Pettis) [25]. En este caso $\langle\cdot,\cdot\rangle$ denota el producto escalar en L^2.

Otro inconveniente de trabajar en espacios de dimensión infinita es que no podemos definir un orden entre los elementos del espacio. Como consecuencia el concepto de la mediana no se puede extender de manera trivial. La mediana, en la estadística multivariante, se define, en términos intuitivos, como el valor que deja a ambos lados tanta distribución de la variable aleatoria. Respectivamente, su estimador para un conjunto de observaciones de esta variable aleatoria, se construye como el elemento en la posición central del conjunto ordenado. Ahora bien, como en el caso funcional no tenemos una manera de "ordenar" nuestros datos ni podemos hablar de "densidades" de un proceso estocástico, necesitamos definir la mediana de otra forma. Esta otra manera apela al siguiente resultado acerca de la mediana en una dimensión:

Teorema 1.11. [13] *La mediana m de una variable aleatoria X cumple la siguiente propiedad:*

$$m = \arg\inf_{c\in\mathbb{R}} \mathbb{E}(|X - c| - |X|).$$

Esta propiedad de la mediana puede utilizarse como una definición alternativa, y será esta la que generalicemos al caso funcional. De esta manera, la mediana (geométrica) de un proceso estocástico se define de la siguiente forma:

Definición 1.12. [14] *Llamamos mediana de un proceso estocástico $M(\mathcal{X})$ al elemento que minimiza la expresión*[4]

$$\varphi(y) = \mathbb{E}(\|\mathcal{X} - y\| - \|\mathcal{X}\|).$$

Gracias a esta propiedad de la mediana, hemos conseguido generalizar este concepto a un espacio funcional.

Además existen otras muchas diferencias entre la estadística multivariante y la estadística con datos funcionales como la no invertibilidad de los operadores de covarianzas [2, c. 12, p. 532].

[4] Gracias a la desigualdad triangular, la mediana siempre está bien definida. Incluso cuando $\mathbb{E}(\|X\|) = \infty$.

Capítulo 2

Clasificación supervisada

En este capítulo comenzaremos definiendo el problema de clasificación supervisada tanto multivariante como funcional (Sección 2.1) para, posteriormente, en la Sección 3.3, explicar las estrategias y metodologías que usaremos a la hora de analizar los datos funcionales en los experimentos que realizaremos en el Capítulo 4.

2.1. Aprendizaje supervisado: Clasificación

El *problema de clasificación* consiste en asignar a una nueva observación de una variable aleatoria un grupo (al que llamaremos clase). Además, le daremos el apellido *supervisada* si, para decidir la clase, contamos con un conjunto de datos clasificados de antemano.

Definición 2.1. [11, p. 1] *Decimos que el par (X,Y) define un problema de clasificación binario si X es una variable aleatoria que toma valores en* $\mathbb{R}^d$ *e Y es una variable aleatoria con rango* $\{0,1\}$.

Para dar una solución a un problema de clasificación necesitamos una manera de asignar a una nueva observación x de X, un valor del rango de Y (clase). Por esto definimos el concepto de solución de la siguiente forma.

Definición 2.2. *Diremos que una solución al problema de clasificación definido por (X,Y) es cualquier función g, medible, que asocie a cada elemento x un único valor del rango de Y.*

A estas soluciones las llamaremos clasificadores, reglas de clasificación o discriminantes.

Nos interesaría poder distinguir entre soluciones "buenas" y "malas". Para poder hacer esto sería conveniente tener una manera de medir el error que comete una solución a este tipo de problemas. Para ello, si tenemos que g es un clasificador, podemos medir el error que comete como

$$L(g) = \mathbb{P}(g(X) \neq Y). \tag{2.1}$$

En otras palabras, el error asociado una regla de clasificación g se corresponde con la probabilidad de que g asocie una nueva observación x la clase equivocada. Con esta definición de *error asociado a un clasificador* estamos dando la misma importancia a todos los tipos de fallos. Por esto, dependiendo del problema que se esté tratando, puede ser conveniente modificar esta definición del error para dar más importancia a una clase que a otra.

Una vez dicho esto, podemos definir la regla de clasificación óptima g^*, conocida como *regla Bayes*, asociada a un problema de clasificación binario como [19, p. 10]

$$g^*(x) = \begin{cases} 1 \text{ si } \eta(x) = \mathbb{E}[Y \mid X = x] = \mathbb{P}(Y = 1 \mid X = x) > \frac{1}{2}, \\ 0 \text{ si no.} \end{cases} \tag{2.2}$$

A continuación vamos a un ejemplo de problema de clasificación supervisada multivariante.

Ejemplo 2. La Forsterita (Mg_2SiO_4), también conocida como olivino canario, es un mineral que se encuentra con frecuencia en zonas volcánicas como las islas de Lanzarote o Gran Canaria (de ahí su nombre). Suele formarse cuando se somete al magnesio a grandes niveles de presión y temperatura

haciendo que cristalice, apareciendo de forma natural con forma de geoda (por fuera parece una simple piedra volcánica pero en su interior alberga el cristal de magnesio). Es por esto por lo que al Instituto Geológico y Minero de España (IGME) le interesa automatizar el proceso de clasificación de este mineral de la siguiente forma:

- Clase 1: Piedras volcánicas con cristal de magnesio en su interior.
- Clase 0: Rocas volcánicas ordinarias.

Para llevar a cabo esta selección nos fijamos en las siguientes variables:

- *Densidad*: La cantidad de magnesio presente en este mineral hace que la densidad de este tipo de rocas sea diferente al resto. Lo medimos en g/cm^3 y lo restringimos al intervalo $[0, 5]$.
- *Conductividad eléctrica*: Una de las propiedades que hace especial al magnesio frente al carbono es su alta conductividad eléctrica. Medimos la conductividad eléctrica en siemens por metro (s/m) y nos restringimos a la región $[0, 10^8]$.

En este caso, X es una variable aleatoria bidimensional y $\Omega = [0, 5] \times [0, 10^8]$ $\frac{g \cdot s}{cm^3 \cdot m}$, y tanto la gráfica de $\eta(x)$ como algunas de sus superficies de nivel pueden verse en la Figura 2.1. Además, en la Figura 2.2, pueden verse dos soluciones a este problema (clasificadores), entre los que se encuentra la *regla Bayes*.

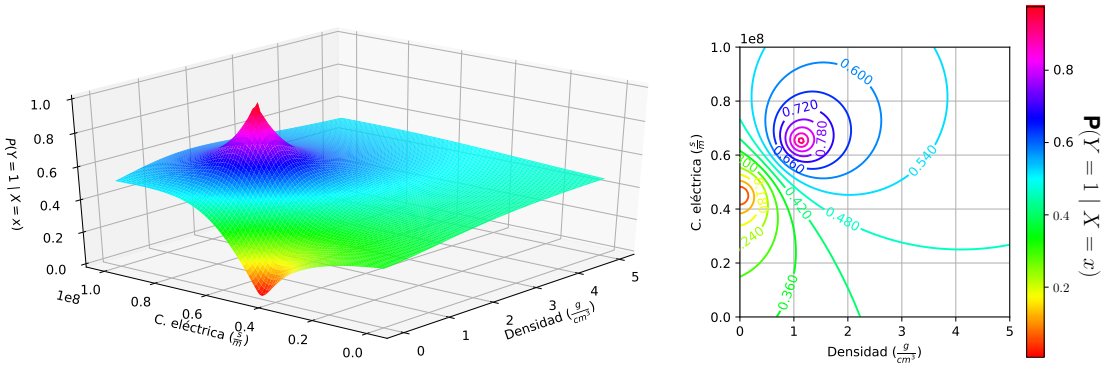

Figura 2.1: A la izquierda, la gráfica de $\eta(x) = \mathbb{E}[Y \mid X] = \mathbb{P}(Y = 1 \mid X = x)$. En la parte derecha sus curvas de nivel.

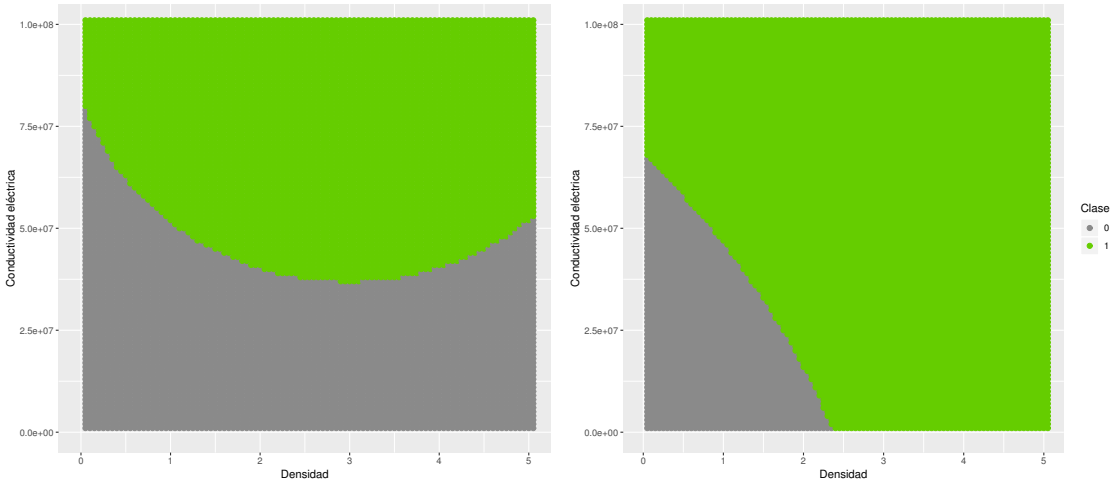

Figura 2.2: A la izquierda el clasificador que le asigna a cada elemento $x \in \Omega$ el valor 1 si $\eta(x) = \mathbb{P}(Y = 1 \mid X = x) > \frac{1}{2}$ (*regla Bayes*), y a la derecha el correspondiente a $\eta(x) > \frac{9}{20}$.

En el Ejemplo 2 los datos solo tienen dos atributos y la covarianza entre las dos variables no tiene una estructura funcional. Sin embargo, cuando utilizamos un enfoque funcional suponemos que X es una discretización de un proceso estocástico $\mathcal{X}$ continuo (y por lo tanto tiene una función de covarianzas continua). Así, definimos el *problema de clasificación funcional* de la siguiente manera.

Definición 2.3. [13] *Decimos que el par $(\mathcal{X}, Y)$ define un problema de clasificación funcional binario si Y es una variable aleatoria con rango $\{0, 1\}$ y $\mathcal{X}$ es un proceso estocástico.*

En el Ejemplo 3 podemos ver un problema en el que los datos son funciones.

Ejemplo 3. A todos nos ha ocurrido alguna vez que escuchamos una canción, sabemos de qué canción se trata, pero no recordamos su título, autor o año en la que fue grabada. Para ayudarnos en este tipo de situaciones existen programas y aplicaciones que son capaces de proporcionarnos todo tipo de información a cerca de una canción con tan solo escuchar un pequeño trozo de esta. Para ello, aplicaciones como Shazam no hacen más que resolver un problema de clasificación en el que los datos que se manejan son funciones. En estas situaciones el problema de clasificación se plantea de la siguiente manera. Dada una pequeña sección de una canción, de unos 5 segundos de duración, se le quiere asignar una clase, que en este caso es su título (una vez obtenido el título de la canción se le podrá otorgar al usuario toda la información que desee con más facilidad). Para modelizar este problema diremos que una onda de sonido es la vibración que ejercen las partículas del aire, y de esta forma podemos ver una canción es un conjunto de vibraciones en el aire a lo largo del tiempo. Para medir estas vibraciones del aire utilizaremos un micrófono, herramienta capaz de medir la presión del aire en diferentes instantes. Así, las canciones que vamos a querer clasificar las estamos viendo como curvas de presión a lo largo del tiempo (funciones). Para grabar una canción utilizaremos un micrófono, herramienta capaz de medir la presión del aire en diferentes instantes. En la Figura 2.3 puede verse una grabación de los primeros 14 segundos de la canción Moondance de Van Morrison.

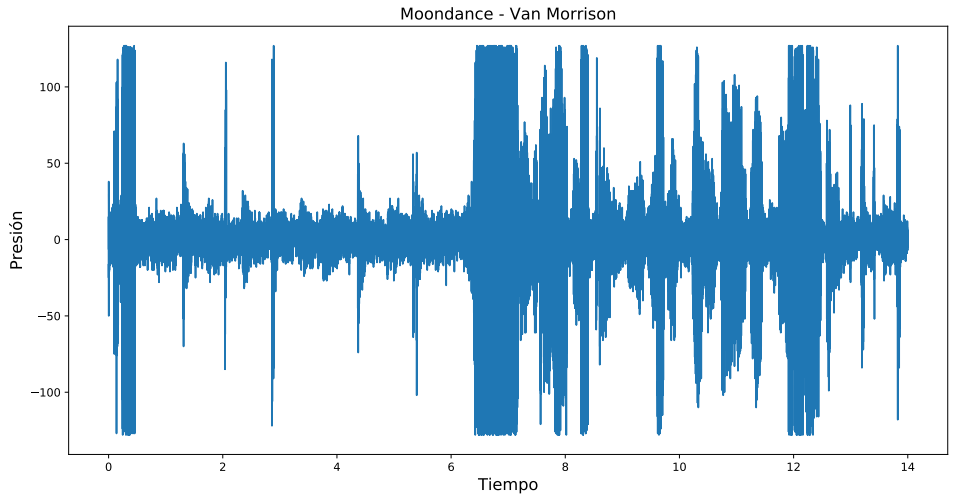

Figura 2.3: Grabación de los primeros 14 segundos de la canción Moondance de Van Morrison. Cada punto de esta gráfica representa la presión del aire (eje y) medida en $10 \log_{10} 20P$ donde P es la presión del aire en micropascales (μPa) a lo largo del tiempo (eje x) medido en segundos. Se ha utilizado una frecuencia de muestreo de 44100 Hz.

Cabe destacar que, aunque este problema de asignar a un trozo de audio el titulo de la canción de la que proviene se puede abordar desde el punto de vista de la clasificación supervisada, el algoritmo de Shazam funciona de manera radicalmente distinta [35].

Cuando nos enfrentamos a un problema funcional los datos que manejamos son mucho más complejos que en los problemas multivariantes. Esto trae consigo propiedades como:

- **Mejor visualización:** Al tratarse los datos de funciones continuas, estas suelen ser más fáciles de representar e interpretar que un dato multivariante.

- **Mayor número de herramientas:** Siempre vamos a suponer que las curvas observadas son funciones continuas, aunque en ocasiones podremos asumir que nuestros datos son además derivables. Esto hace que podamos aplicar a nuestros datos una mayor variedad de herramientas de procesamiento y análisis que en el caso multivariante. En muchos problemas podremos derivar e integrar las funciones, representarlas en bases (de Fourier, splines, ...) o aplicar técnicas más sofisticadas como un análisis del tiempo y la frecuencia si trabajamos con señales acústicas (en la Figura 2.4 puede verse el análisis del tiempo y la frecuencia (espectrograma) de la grabación 2.3).

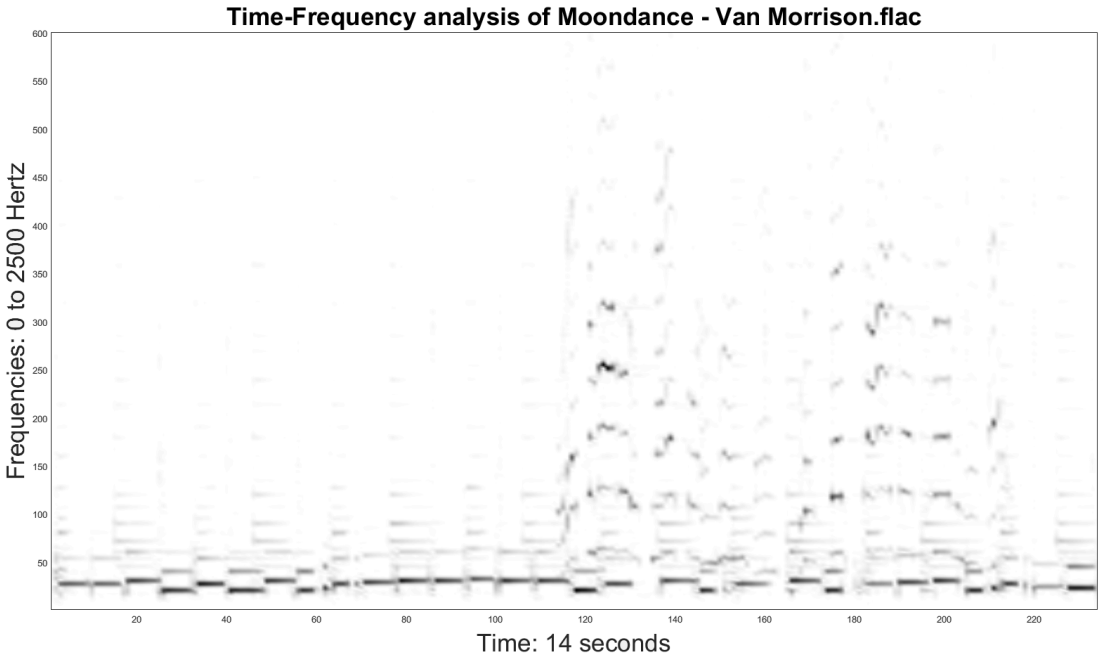

Figura 2.4: En este espectrograma se pueden ver las intensidades de cada frecuencia en cada instante de la grabación del Ejemplo 3. Esta transformación de la señal original la utiliza la aplicación Shazam para identificar canciones [35].

Pasamos ahora a enunciar uno de los teoremas más importantes de esta teoría. El Teorema 2.4, asegura que el clasificador de Bayes es el clasificador óptimo en el sentido de que es la regla de clasificación que tiene menor error asociado, es decir, que se equivoca con menor probabilidad.

Teorema 2.4. [21, c. 10, p. 262] *El error asociado a cualquier clasificador g es mayor o igual que el error asociado a la regla Bayes.*

$$L(g^*) \leq L(g). \tag{2.3}$$

La demostración de este teorema puede consultarse en el Apéndice 5.1. Con este resultado vemos que el clasificador de Bayes es el óptimo. Ahora bien, en ocasiones trabajaremos con soluciones distintas a la óptima y nos puede interesar cuan lejos estamos de la mejor solución. Normalmente no se conoce la distribución de $Y \mid X = x$ (y por lo tanto tampoco la probabilidad $\mathbb{P}(Y = 1 \mid X = x) = \mathbb{E}[Y \mid X = x]$) y si encima suponemos que X es un proceso estocástico en lugar de un vector aleatorio, ni siquiera podemos hablar de densidad. Es por esto por lo que, para buscar clasificadores buenos, existen principalmente dos enfoques distintos [19, c. 1, p.15]:

- *plug-in*: Como conocer la función $\eta(x)$ nos aporta mucha información a cerca del comportamiento de los datos, este enfoque radica en estimar la función de regresión.
- *Minimización el riesgo empírico*: Otra manera de crear soluciones es viendo cómo estas soluciones clasifican los datos de nuestra muestra de entrenamiento (con datos previamente clasificados). Para ello se sustituye la definición de *error* por su correspondiente versión muestral y se entrena el clasificador minimizando dicho error.

Definición 2.5. *Dada una muestra de entrenamiento*

$$D_n = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}; \quad \text{con } x_i \in \mathbb{R}^d; \quad y_i \in \{0, 1\},$$

donde n es el tamaño de la muestra, d la dimensión del vector de atributos e y_i la clase asociada al elemento x_i de la muestra, definimos el error empírico o error muestral como

$$L_{D_n}(g(x)) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{g(x_i) \neq y_i\}}, \tag{2.4}$$

Esto es, la proporción de veces que el discriminante clasifica mal los elementos de D_n.

La mayor desventaja del error empírico es que se utiliza el mismo conjunto de datos para crear el clasificador y para comprobar su eficacia. Para reducir este sesgo suelen utilizarse técnicas como *validación cruzada* [27, c. 5, p. 176] o *bootstrap* [27, c. 5, p. 187].

Capítulo 3

Metodologías

En este capítulo vamos a introducir los algoritmos de aprendizaje supervisado y las herramientas de procesamiento de datos que vamos a utilizar en el Capítulo 4 para generar clasificadores a partir de un conjunto de datos cuando apliquemos cualquiera de los dos enfoques. Comenzamos explicando el funcionamiento del algoritmo Random Forest (RF) y de las Máquinas de Vector Soporte (SVM) y viendo algunos ejemplos de cómo estos métodos se han aplicados a problemas de clasificación supervisada funcionales. Por último explicaremos las técnicas de procesamiento y análisis de datos que utilizan propiedades funcionales que usaremos en el Capítulo 4.

3.1. Random Forest

El algoritmo de Random Forest tiene sus orígenes en el año 1995, año en el cual, la matemática surcoreana Tin Kam Ho publica el artículo *Random Decision Forests* [26]. En este artículo se presenta por primera vez la metodología de combinado de clasificadores. Leo Breiman se basará en esta metodología para definir el aglgoritmo *Random Forest*, en el cual, cada árbol del bosque genera un clasificador en un subespacio aleatorio del espacio de características. Por último se aplica a un problema de reconocimiento de dígitos manuscritos.

> The essence of the method is to build multiple trees in randomly selected subspaces of the feature space. Trees in, different subspaces generalize their classification in complementary ways, and their combined classification can be monotonically improved. The validity of the method is demonstrated through experiments on the recognition of handwritten digits.
>
> T. K. Ho - *Random Decision Forests* - Abstract - [26]

Aunque este algoritmo no se popularizó hasta que, en el año 2001, Leo Breiman publica el famoso artículo *Random Forests* [8], en el cual da una definición rigurosa del algoritmo y demuestra algunas propiedades, como la velocidad de convergencia o cotas del error. Además lo aplica a un problema de clasificación con múltiples clases, y muestra como se podría aplicar una variación de esta metodología para dar soluciones a problemas de regresión.

Tras la publicación de Breiman, el algoritmo RF comenzó a aplicarse en prácticamente todas las ramas de la ciencia para resolver problemas tanto de clasificación como de regresión. Algunos ejemplos de la aplicación de Random Forest pueden verse en los artículos [1], en el cual Alam y Vuong lo aplican para detectar posibles malwares en aplicaciones de dispositivos Android, [33], donde Shi, Seilgson y otros lo usan para clasificar tumores en benignos y malignos, o en [6], artículo en el cual Bosch, Zisserman y Muñoz dan soluciones a un problema de clasificación de imágenes. Cabe destacar la gran popularidad de la que goza Random Forest en el mundo de la biología. Por lo general, los problemas de clasificación propios del ámbito de la biología, destacan por tener conjuntos de datos en los que el número de variables es mucho mayor que la cantidad de datos. Esto se debe a que el Random Forest implícitamente realiza un proceso de selección de variables.

Desde el punto de vista teórico, pocos han sido los resultados demostrados a partir de la publicación del artículo de Breiman en 2001, sin embargo, sí que se han hecho algunos avances. Entre ellos destacan los resultados propuestos por Biau en [4] en 2012, por Biau, Devroye y Lugosi en [5] en 2008 y por Wagner en [34] en 2014. En el primero Biau prueba que la velocidad de convergencia

del algoritmo sólo depende del número de atributos que de verdad explican el comportamiento de los datos, y no de los atributos ruidosos. En el segundo, Biau, Devroye y Lugosi se amplían los resultados obtenidos por Breiman en su artículo de 2001 y se profundiza en la aplicación de Random Forests en problemas de regresión. Por último Wagner prueba en 2014 un resultado asintótico a cerca de la velocidad de convergencia de Random Forests.

Desde que se definió este algoritmo en 1995, prácticamente todas sus aplicaciones vienen de problemas de clasificación supervisada con datos multivariantes, y en muy escasas ocasiones se ha tratado de aplicar este algoritmo en problemas que involucren datos funcionales. De hecho, la estrategia más usada para atacar problemas funcionales con este algoritmo, reside en transformar los datos funcionales en multivariantes, ignorando así gran cantidad de información funcional. Un claro ejemplo de esta estrategia puede verse en el artículo de Rahman, Drhuba y otros [28] de 2019, en el cual se transforman los datos funcionales en multivariantes promediando las funciones en diferentes intervalos, y, tras esto, aplicando el algoritmo de Random Forest clásico al nuevo conjunto de datos.

El método de Random Forest (RF) es un algoritmo de aprendizaje automático que se utiliza para solucionar problemas de clasificación supervisada y regresión. Este método se basa en la combinación de árboles predictores entrenados mediante una muestra bootstrap del conjunto de datos de entrenamiento, donde cada árbol es un clasificador con las fronteras de decisión paralelas a los ejes. Para generar los árboles predictores se pueden utilizar varios algoritmos, como ID3 o C4.5, aunque el más común, y el que nosotros usaremos, es el CART. El algoritmo CART es un procedimiento de división del espacio que funciona de forma iterativa y que genera fronteras según un conjunto de datos. Para buscar una buena división del espacio CART busca la región con fronteras paralelas a los ejes que maximice la *ganancia de pureza* de las nuevas regiones, donde R es la región que queremos dividir y $R_{1;i,\xi}$ y $R_{2;i,\xi}$ son el resultado de separar R por la variable i por el valor ξ

$$R \longrightarrow \underbrace{\left[R \cap \{x \in \mathbb{R}^d : \ x_i < \xi\}\right]}_{R_{1;i,\xi}} \cup \underbrace{\left[R \cap \{x \in \mathbb{R}^d : \ x_i \geq \xi\}\right]}_{R_{2;i,\xi}}. \tag{3.1}$$

Además se define la ganancia de pureza que resulta de dividir R en $R_{1;i,\xi}$ y $R_{2;i,\xi}$ como

$$G(R; i, \xi) = I(R) - \underbrace{\left(\frac{|R_{1;i,\xi}|}{|R|} I(R_{1;i,\xi}) + \frac{|R_{2;i,\xi}|}{|R|} I(R_{2;i,\xi})\right)}_{\text{suma de impurezas ponderadas de las nuevas regiones}}, \tag{3.2}$$

donde $I(R)$ denota la impureza de la región R, la cual se define como $I(R) = \psi(s, 1-s)$ con s la proporción de elementos de la clase 1 en R y ψ es una función que cumple las propiedades

$$\psi\left(1/2, 1/2\right) \geq \psi(s, 1-s) \quad \forall s \in [0, 1]. \tag{3.3}$$

$$\psi(0, 1) = \psi(1, 0) = 0. \tag{3.4}$$

$$\psi(s, 1-s) \text{ es creciente en } s \in [0, 1/2] \text{ y decreciente en } s \in [1/2, 1]. \tag{3.5}$$

En este trabajo utilizaremos la función de impureza de Gini $\psi(s, 1-s) = 2s(1-s)$. Dividiendo las regiones de forma iterativa, CART es capaz de entrenar clasificadores (árboles) dada una muestra bootstrap B^k. Una particularidad de los árboles predictores es que, en lugar de buscar la mejor separación entre todas las variables de los datos, solo se fija en $\boldsymbol{\nu}$ de ellas (en la librería sklearn este parámetro se le denomina `max_features`). Para obtener el clasificador Random Forest $G(x)$ a partir de los árboles agrupamos mediante una "votación por mayoría" siguiendo la fórmula

$$G(x) = \begin{cases} 1 \text{ si } \sum_{k=1}^{\boldsymbol{\tau}} g_k(x, B^k) \geq \sum_{k=1}^{\boldsymbol{\tau}} 1 - g_k(x, B^k), \\ 0 \text{ si no,} \end{cases}$$

donde $\boldsymbol{\tau}$ es el número de árboles del bosque (recibe el nombre `n_estimators` en la librería sklearn) y $g_k(x, B^k)$ es un árbol generado por el algoritmo CART y entrenado a partir de una muestra bootstrap B^k.

La esencia de este método radica en promediar muchos modelos simples (árboles) para obtener un clasificador más complejo. Random Forest destaca frente a otros algoritmos por ser capaz de manejar conjuntos de datos de alta dimensionalidad. Esto se debe a que la búsqueda de particiones óptimas se realiza usando un subconjunto de atributos de dimensión menor que el original.

3.2. Máquinas de Vector Soporte

Las Máquinas de Vector Soporte tienen su origen en el año 1995, año en el que Corinna Cortes y Vladimir Vapnik publican el artículo *Support Vector Networks* [12], en el cual proponen las SVM para generar soluciones a problemas de clasificación binarios.

> The *support-vector network* is a new learning machine for two-group classification problems. (...) The idea behind the support-vector network was previously implemenTeed for the restricted case where the training data can be separated without errors. We here extend this results to non-separable training data.
>
> Cortes. C & Vapnik. V - *Support-Vector Network* - Abstract - [12]

Las SVM se diseñaron para solucionar problemas de clasificación multivariante, sin embargo, también ha sido utilizado en diversas ocasiones cuando los problemas que se pretenden analizar se modelizan con un enfoque funcional. Un ejemplo se puede encontrar en el artículo de Rossi y Villa de 2006 [30]. Aquí se muestra cómo usar esta familia de algoritmos puede resultar realmente útil cuando se trabaja con este enfoque, y se prueban diferentes núcleos en función del problema que se esté estudiando. Tras aplicar las SVM a una serie de conjuntos de datos funcionales, los autores llegan a la conclusión de que son de bastante precisos cuando se aplica a problemas de naturaleza funcional.

Otro ejemplo se puede ver en el artículo de Cai, Han y otros de 2003 [9], en el cual se aplican diferentes SVM para clasificar proteínas en clases dependiendo de su propósito. Para ello los autores realizan un preprocesamiento de los datos mediante un análisis de componentes principales para después entrenar varias máquinas de vector soporte con distintos núcleos. De esta forma se obtiene una tasa de acierto promedio de cerca del 90 %, mejorando así los clasificadores creados hasta entonces para resolver este problema. Un caso similar a este puede encontrarse en su artículo de 2003 [10]. Aquí se pretende predecir el propósito de diferentes conjuntos de compuestos de pseudo aminoácidos. Para ello, se formula el correspondiente problema de clasificación supervisada aplicando un enfoque funcional. Tras esto se implementan varias SVM con diferentes núcleos y se muestran los resultados y conclusiones obtenidas, siendo estos "satisfactorios".

Por último, en el artículo de 2014 de Xue, Du y Su [36], puede verse cómo se implementan unas SVM a un conjunto de datos funcionales que han sido preprocesados mediante un análisis hiperespectral (esto es, se han representado las imágenes originales en la base de Fourier dos dimensional). De esta forma se consigue realizar un proceso de selección de variables (gracias al análisis hiperespectral), y aplicando las SVM al conjunto de datos transformado, se logra generar clasificadores. Aquí puede verse cómo pasar al espacio de fases es una técnica ha tener en cuenta al trabajar con datos funcionales.

Para explicar el fundamento teórico de las SVM recurriremos al libro *Learning with kernels* de Schölkopf y Smola [32, c. 2, p. 187]. El método de las Máquinas de Vector Soporte (SVM) hacer referencia a una familia de algoritmos de aprendizaje supervisado que son aplicados tanto a problemas de clasificación supervisada como de regresión. Para generar un clasificador, las SVM buscan un hiperplano (o conjunto de hiperplanos) que separe los datos de las dos clases y dejando un margen lo más grande posible. Si suponemos que el problema es linealmente separable y llamamos H a un hiperplano que se pare las dos clases, se tiene que el margen que se busca maximizar vale

$$Margen = \frac{1}{\|w\|}, \tag{3.6}$$

donde w es un vector normal a H. Este margen lo ajustaremos más adelante con el parámetro `C` de la función `svm.SVC` de la librería `sklearn`. Para buscar el mejor hiperplano usamos la técnica de los multiplicadores de Lagrange. Con esto buscamos los valores b y w que minimizan la expresión

$$L = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i (w \cdot x_i^{SV,1} + b) + \alpha_i + \sum_i \alpha_i (w \cdot x_i^{SV,0} + b) + \alpha_i, \tag{3.7}$$

donde $x_i^{SV,0}$ y $x_i^{SV,1}$ denotan los vectores soporte i-ésimos de las clases 0 y 1 y el símbolo $\cdot$ representa el producto interno usual en $\mathbb{R}^d$. Si derivamos L con respecto a w y b e igualamos a 0 los resultados llegamos a que el vector normal w tiene la fórmula

$$w = \sum_i \alpha_i x_i^{SV,1} - \sum_i \alpha_i x_i^{SV,0}. \tag{3.8}$$

Es decir, w se escribe mediante una combinación lineal de los vectores soporte. Además, si definimos

$$sign(x_i^{SV}) = \begin{cases} 1 \text{ si } x_i^{SV} \text{ pertenece a la clase 1,} \\ -1 \text{ si no,} \end{cases} \tag{3.9}$$

entonces tenemos que los vectores soporte han de satisfacer

$$\sum_i \alpha_i sign(x_i^{SV}) = 0. \tag{3.10}$$

Con esta notación podemos reescribir L como

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j sign(x_i^{SV}) sign(x_j^{SV}) x_i^{SV} \cdot x_j^{SV}. \tag{3.11}$$

Así obtenemos la siguiente regla de clasificación

$$g(u) = \begin{cases} 1 \text{ si } \sum_i \alpha_i sign(x_i^{SV}) x_i^{SV} \cdot u + b \geq 0, \\ 0 \text{ si no.} \end{cases} \tag{3.12}$$

Lo sorprendente de este método es que, como se puede ver en las ecuaciones 3.11 y 3.12, tanto la regla de decisión como la expresión L dependen únicamente del producto escalar entre los vectores soporte y el nuevo dato a clasificar. Habrá problemas en los que los datos no sean linealmente separables. En estos casos aplicamos lo que se conoce como "kernel trick". Este truco consiste en transformar los datos en un espacio de dimensión superior (o incluso infinita) donde es más probable que sí se puedan separar por un hiperplano. Para ello utilizamos las *kernel functions* o *funciones núcleo* (o simplemente núcleos). Estas funciones se utilizan para calcular el producto interno entre dos elementos directamente en el espacio transformado. Se suelen utilizar funciones núcleo como

$$\text{Polinomial homogéneo: } k(x_i, x_j) = (x_i \cdot x_j)^d, \tag{3.13}$$

$$\text{Tangente hiperbólica: } k(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c) \text{ para algunos } \kappa > 0 \text{ y } c < 0 \tag{3.14}$$

$$\text{RBF gaussiano: } k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \text{ con } \gamma > 0, \tag{3.15}$$

aunque nosotros solo utilizaremos los núcleos polinomial homogéneo con $d = 1$ (lineal) y RBF gaussiano. El hecho de que las SVM transformen los datos llevándolos a espacios de dimensión superior hace que sean bastante útiles en problemas de alta dimensionalidad, aunque suelen ser muy costosas en cuanto a tiempos de ejecución.

3.3. Procesamiento de datos

En un problema de clasificación funcional los datos de los que disponemos son funciones. Sin embargo, dado que no podemos almacenar una cantidad infinita de valores, trabajaremos con una discretización: en lugar de almacenar la función $x(t)$ trabajamos en la discretización $\{(t_i, x(t_i))\}_{i=0}^{N-1}$. La forma más sencilla de recuperar el dato funcional original $x(t)$ consiste en interpolar los puntos de la discretización de manera lineal, mediante la fórmula

$$\widehat{x(t)} = \frac{x(t_i)(t_{i+1} - t) + x(t_{i+1})(t - t_i)}{t_{i+1} - t_i} \quad \text{para } t_i \leq t \leq t_{i+1}. \tag{3.16}$$

Esta es la forma más precisa de aproximar nuestro dato si solo conocemos los valores en la discretización. A la hora de estudiar los datos funcionales nos interesará conocer información más allá de sus valores puntuales. Para ello procesaremos los datos funcionales aplicando mediante una serie de estrategias que mostramos a continuación:

- **Sin procesar**: La primera estrategia que vamos a seguir es la de estudiar el comportamiento de los valores de las funciones en diferentes instantes de tiempo. Esta metodología no tiene en cuenta la naturaleza funcional de los datos. No obstante es una manera de afrontar un problema de clasificación a tener en cuenta. En este caso llamaremos conjunto de datos original (sin procesar, en bruto, en crudo ...) a los valores de las funciones medidos en los diferentes instantes temporales, y será este conjunto el que utilicemos para entrenar los Random Forest y las SVM.

- **Componentes principales**: La segunda forma de afrontar el problema de clasificación funcional es mediante una selección de las componentes principales del conjunto de datos original (el que consta de los valores de las trayectorias en los instantes temporales en cuestión). Hacemos un análisis de la sensibilidad del acierto para los diferentes clasificadores para seleccionar, en función del algoritmo, una cantidad de componentes principales razonable. Una vez se ha escogido una cantidad de componentes principales se entrenan los clasificadores con este nuevo conjunto de datos. A este nuevo conjunto le llamamos *conjunto de las componentes principales*. Esta estrategia constituye un enfoque multivariante para resolver el problema, ya que al quedarnos con una cantidad de componentes principales estamos perdiendo cualquier estructura funcional (de continuidad) que se pudiera tener.

- **Proyección en la base de Fourier**: Esta tercera estrategia es la primera que utiliza un enfoque funcional para dar una solución al problema de clasificación. Proyectamos las trayectorias del conjunto original en la base de Fourier truncada con n_F elementos

$$\{e^{2\pi inx}\} \text{ con } \begin{cases} \lfloor -\frac{n_F}{2} \rfloor \leq n \leq \lfloor \frac{n_F}{2} \rfloor \text{ si } n \text{ es impar,} \\ \lfloor -\frac{n_F - 1}{2} \rfloor \leq n \leq \frac{n_F}{2} \text{ si } n \text{ es par,} \end{cases} \tag{3.17}$$

, donde n_F es un parámetro fijado de antemano, y consideramos sus coordenadas (coeficientes de Fourier) en esta base. Para ello usamos la fórmula de la Definición 1.4. Cabe destacar que para cada valor de n_F tendremos un total de $2n_F + 1$ coeficientes de Fourier. Esto se debe a que el coeficiente correspondiente a $n = 0$ siempre es un número real, mientras que el resto de coeficientes pueden ser complejos. En este caso nos quedamos con sus coordenadas real e imaginaria usando la fórmula de De Euler [23, c. 1 p. 6]

$$z = r \cdot e^{i\theta} = r\cos(\theta) + ir\sin(\theta) \tag{3.18}$$

para poder trabajar con números reales. Así podemos codificar la función proyectada usando sus coeficientes en la base de Fourier. Llamamos *conjunto de datos proyectados en la base de Fourier* o simplemente *conjunto de datos proyectados* a estas partes reales e imaginarias y lo utilizamos para entrenar los clasificadores. Tomamos el valor óptimo de n_F mediante validación cruzada para cada clasificador (puede que no valga el mismo n_F para entrenar un RF y una SVM). En la Figura 3.1 se ve el resultado de proyectar una función en la base de Fourier con dos valores de n_F diferentes.

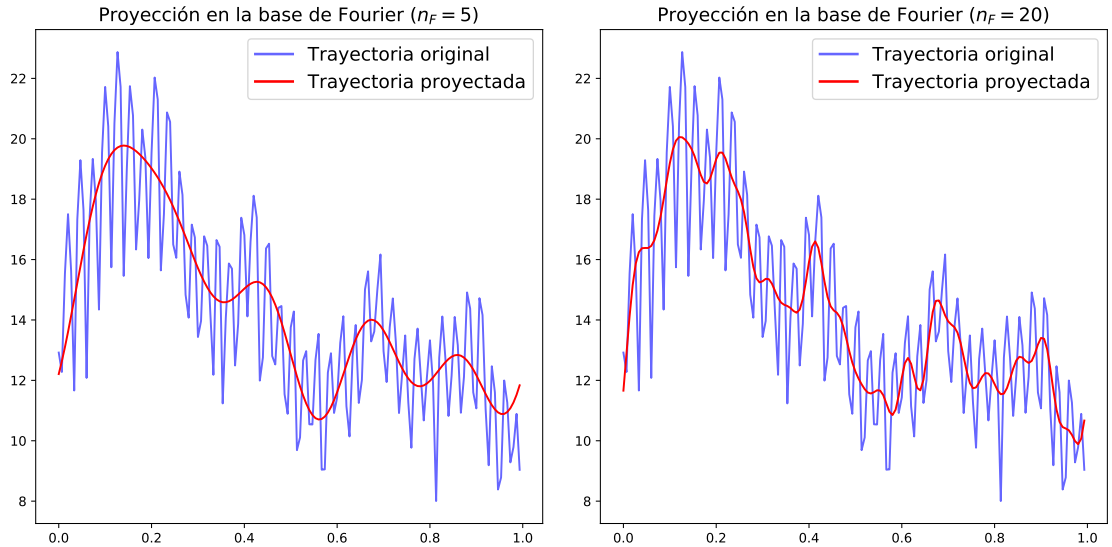

Figura 3.1: Proyección de una trayectoria del conjunto de datos `Phoneme` en la base de Fourier con $n_F = 5$ (izquierda) y $n_F = 20$ (derecha).

- **Agrupado**: Esta estrategia consiste en combinar las componentes principales halladas mediante la segunda estrategia y los coeficientes de Fourier obtenidos mediante la metodología

anterior para entrenar los clasificadores. De esta manera se aplica un enfoque que por un lado es multivariante (ya que se trabaja con las componentes principales) y que por otro es funcional (a la hora de usar los coeficientes de Fourier de las trayectorias). Por esto decimos que este es un enfoque híbrido. Al conjunto de datos formado por las componentes principales y los coeficientes de Fourier le llamamos *conjunto agrupado*.

- **Troceado**: La quinta estrategia que usaremos consiste en hacer que los clasificadores se entrenen centrándose en los datos funcionales restringidos a un intervalo $[n, \ldots, m]$. Para seleccionar una buena pareja de parámetros n y m hacemos un análisis de la sensibilidad del acierto utilizando Random Forest. Para cada pareja de valores de n y m en una malla fijada de antemano entrenamos una serie de Random Forest con los datos restringidos al intervalo $[n, \ldots, m]$. Aquella pareja de parámetros con la que se consigan mejores resultados será la seleccionada para generar el *conjunto de datos troceado* y entrenar el resto de clasificadores. Esta estrategia puede verse como una selección de variables en el que las variables que se escogen han de ser consecutivas. De esta forma se aprovecha la propiedad de la continuidad de los datos que manejamos. Es por esto por lo que consideramos este enfoque como funcional (aprovecha la propiedad de continuidad de los datos). En la Figura 3.2 podemos ver un ejemplo de una trayectoria troceada.

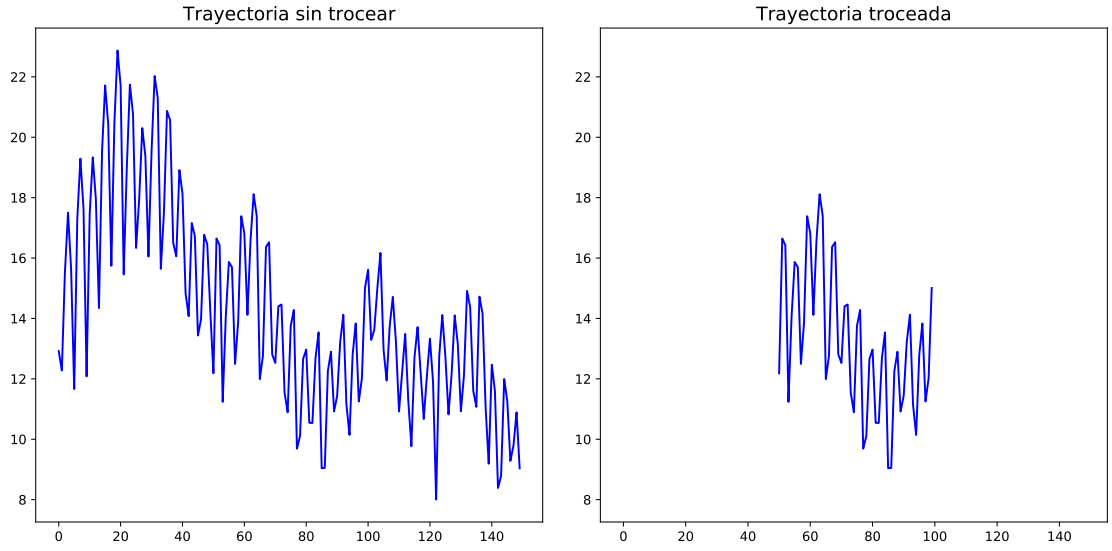

Figura 3.2: Trayectoria del conjunto de datos sin procesar (izquierda) y esta misma curva restringida al intervalo $[50, \ldots, 100]$ (derecha).

- **Selección de variables**: La última estrategia que usaremos será una selección de variables. Las variables que se van a seleccionar son los instantes de tiempo en los que se evalúan los datos funcionales. Además escogemos la cantidad de variables a seleccionar mediante validación cruzada para cada familia de clasificadores. De esta manera, para cada clasificador, generamos el conjunto *conjunto de selección de variables* que usaremos para entrenar cada clasificador.

Capítulo 4

Experimentos y resultados

En este capítulo vamos a analizar las diferencias entre los clasificadores obtenidos a partir de un procesamiento funcional (que utilice técnicas que tengan en cuenta la naturaleza o propiedades funcionales) de los conjuntos de datos y uno multivariante. Para ello realizamos tres experimentos, a los que llamaremos `Brownianos`, `Berkeley` y `Phoneme`. En los tres tendremos un conjunto de datos funcionales (al que llamaremos conjunto original) y lo procesaremos utilizando las técnicas propuestas en la Sección 3.3 (algunas aprovechan las propiedades funcionales y otras no). Después entrenaremos una serie de clasificadores Random Forest y SVM con núcleos lineal y RBF para ver que técnica de procesamiento obtiene mejores resultados.

Una vez tengamos las modificaciones del conjunto original entrenamos una serie de Random Forest, SVMs con núcleo lineal y SVMs con núcleo RBF. Cada uno de estos métodos requiere seleccionar unos hiperparámetros (`max_features` y `n_estimators` en el caso de Random Forest, `C` en el de las SVM con núcleo lineal y `C` y γ en el caso de las SVM con núcleo RBF). Para ello hacemos una partición entrenamiento/test (la partición de entrenamiento contiene $2/3$ de la cantidad total de los datos y la de test $1/3$). Para escoger los mejores valores de los hiperparámetros hacemos una validación cruzada con 10 grupos (10-CV) en el conjunto de entrenamiento, usando las mallas $\{10^{-8}, 10^{-7}, \ldots, 10^{7}, 10^{8}\}$ para `C` y γ. Para ello usamos la función `GridSearchCV` de la librería `sklearn`. Así computamos los aciertos de validación cruzada, test y los tiempos de ejecución. En el caso de los hiperparámetros `n_estimators` y `max_features` de Random Forest hacemos un análisis de la saturación del acierto.

Para poder afirmar que los resultados obtenidos con una familia de clasificadores son significativamente diferentes a los de otra, realizaremos una serie de contrastes de hipótesis mediante el test de Wilcoxon, como sugiere Janez Demšar en su artículo *Statistical comparisons of classifiers over multiple data sets* [18]. Usamos este test para comparar las tasas de acierto promedio de los clasificadores ya que no asume normalidad ni homogeneidad en las varianzas. A lo largo de todo el trabajo, cuando usemos el test de Wilcoxon para comparar las tasas de acierto de dos métodos (digamos A y B), tomaremos como hipótesis nula e hipótesis alternativa

$$H_0: \quad \mathbb{P}(\alpha_0 > \alpha_1) = \mathbb{P}(\alpha_1 > \alpha_0); \qquad H_a: \quad \mathbb{P}(\alpha_0 > \alpha_1) \neq \mathbb{P}(\alpha_1 > \alpha_0), \tag{4.1}$$

donde α_0 es una observación del acierto de un clasificador entrenado con el método A y α_1 con el método B.

4.1. Brownianos con distinta esperanza

El primer experimento que vamos a realizar viene dado por unos conjuntos de datos sintéticos. Cada clase constará de 500 observaciones discretizadas en un total de 100 instantes temporales, de procesos brownianos con diferentes esperanzas e idéntica varianza. Las trayectorias de la clase 0 son observaciones de un proceso browniano estándar mientras que las de la clase 1 se corresponden con observaciones de un proceso browniano con esperanza

$$\mathbb{E}[\mathcal{X} \mid \mathbf{Y} = 1] = \sin(2\pi t/100) \cdot a,$$

donde a es un parámetro que fijamos de antemano para controlar la dificultad del problema. a mide la intensidad de la señal frente al ruido. En concreto tomaremos los valores de 1, 0.5 y 0.1.

Comenzamos haciendo un análisis descriptivo de los datos. Para ello graficamos las medias y desviaciones típicas de los datos junto con algunas trayectorias para los diferentes valores de a. Estas gráficas pueden verse en las Figuras 4.1, 4.2 y 4.3.

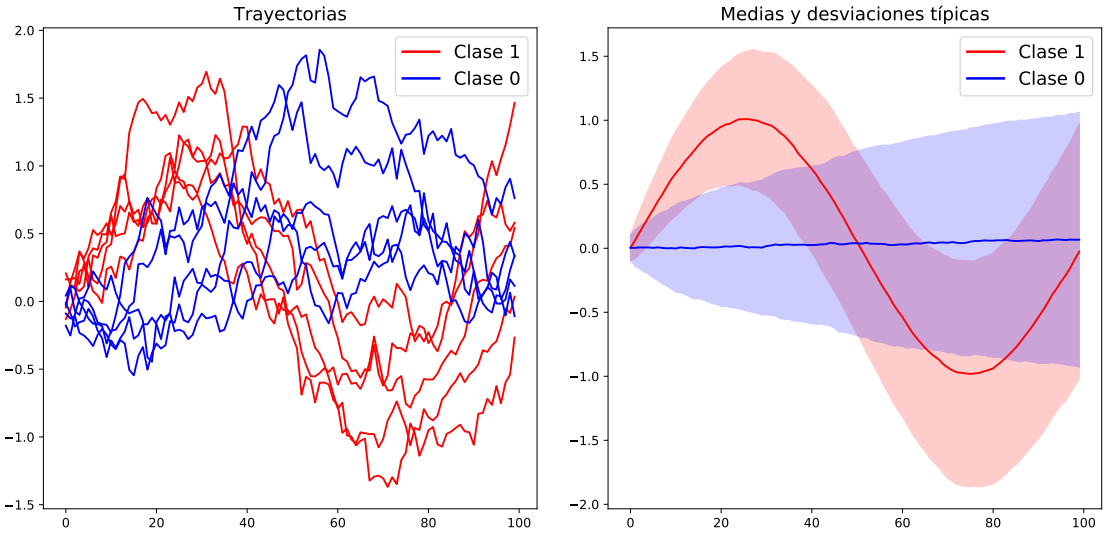

Figura 4.1: A la izquierda 5 trayectorias de cada una de las clases para $a = 1$. A la derecha, las medias y desviaciones típicas.

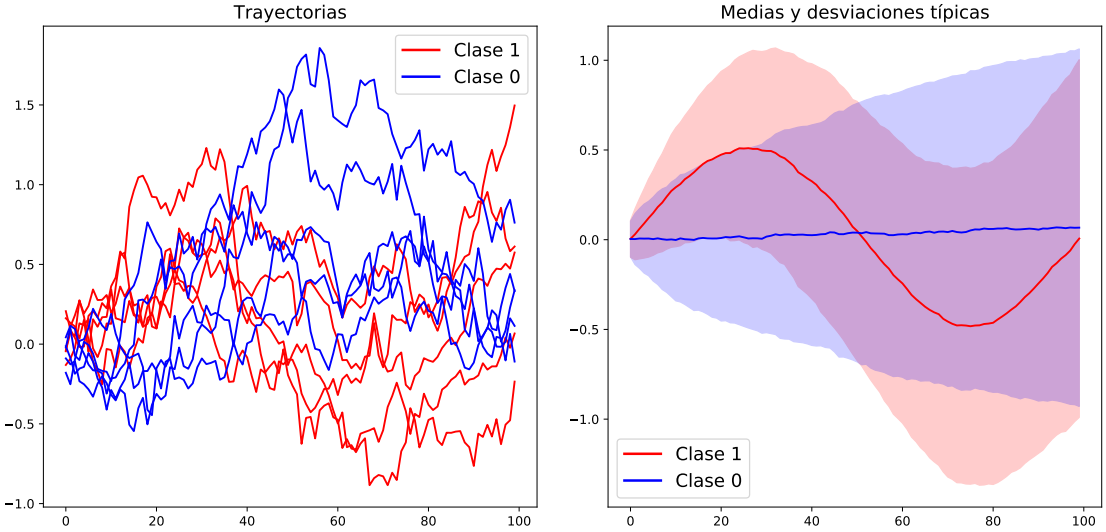

Figura 4.2: A la izquierda 5 trayectorias de cada una de las clases para $a = 0{,}5$. A la derecha, las medias y desviaciones típicas.

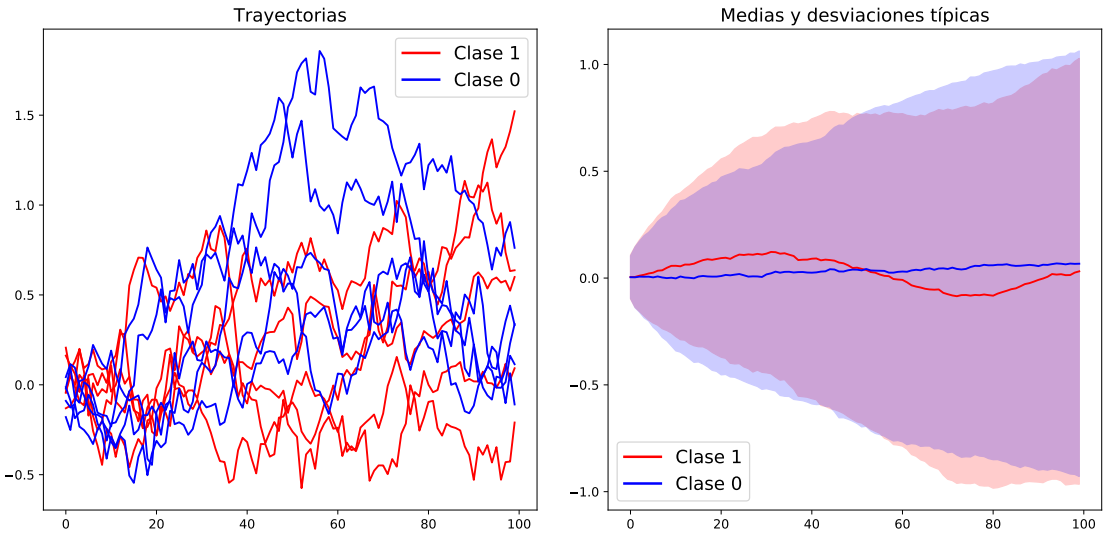

Figura 4.3: A la izquierda 5 trayectorias de cada una de las clases para $a = 0{,}1$. A la derecha, las medias y desviaciones típicas.

Como puede verse en la Figura 4.3 el problema para $a = 0{,}1$ será difícil de resolver con cualquiera de los dos enfoques. Por otro lado, para $a = 0{,}5$ y $a = 1$ se deberían obtener clasificadores con tasas de acierto altas.

Comenzamos con el conjunto de datos sin procesar (original). El primer algoritmo que vamos a utilizar es Random Forest. Para seleccionar los hiperparámetros `max_features` y `n_estimators` realizamos un estudio de la sensibilidad de la tasa de acierto. Fijamos el número de atributos `max_features` en 10 (la raíz cuadrada de la cantidad total de atributos suele ser un valor razonable) y entrenamos 50 RF con diferentes particiones entrenamiento/test para los valores de `n_estimators` en la malla $\{1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047\}$. En la Figura 4.4 podemos ver cómo varía la tasa de acierto acierto de test en función de los valores de `n_estimators`. En los tres conjuntos de datos se puede ver cómo el acierto aumenta según lo hace el número de árboles del RF hasta aproximarse al límite asintótico. Este patrón se repetirá a lo largo de los experimentos. Tomamos 1023 árboles en los tres problemas. En estos valores el acierto se ha saturado y ha dejado de aumentar. Una vez seleccionado el número de árboles del RF vemos si la elección de tomar la raíz cuadrada del número total de atributos (10) es una elección razonable para el hiperparámetro `max_features`. Para ello volvemos a entrenar 50 Random Forest tomando diferentes cantidades del hiperparámetro. Utilizamos la malla $\{1, 2, \ldots, 29\}$. Los resultados pueden verse en la Figura 4.4. Puede verse cómo el parámetro `max_features` no parece influir significativamente en el acierto. Podríamos tomar un valor más pequeño para este hiperparámetro, pero seguimos escogiendo 10 por ser consistentes con la elección estándar (para todos los valores de a). A lo largo de los experimentos, salvo que se especifique lo contrario, tomaremos como valor de `max_features` la raíz cuadrada del número total de atributos.

Con estas configuraciones de los hiperparámetros de Random Forest entrenamos un nuevo clasificador con los conjuntos de entrenamiento. Obtenemos unas tasas de acierto en test del 92.9 % ($a = 1$), 79.1 % ($a = 0{,}5$) y 53.3 % ($a = 0{,}1$). Se han requerido unos tiempos de ejecución de 0.98, 0.93 y 0.97 segundos respectivamente. Como cabía esperar los aciertos disminuyen según lo hace a. Esto se debe a que los conjuntos de datos son más fáciles de separar cuanto mayor es a. No obstante, se obtienen unos resultados relativamente bajos comparados con los demás clasificadores que vamos a entrenar y metodologías que vamos a usar. Por otro lado, los tiempos de ejecución empleados para entrenar el algoritmo son muy bajos.

A continuación entrenamos una SVM con núcleo lineal. Obtenemos el valor óptimo del hiperparámetro `C` mediante una 10 validación cruzada. Los valores obtenidos han sido `C`$= 10^{-4}$ con $a = 1$, `C`$= 10^{-3}$ en el problema con $a = 0{,}5$ y `C`$= 10^{-2}$ en el último problema ($a = 0{,}1$). Conseguimos unos aciertos en validación cruzada y test de 94.6 % y 95.0 % ($a = 1$), 78.7 % y 79.1 % ($a = 0{,}5$) y 55.3 % y 55.7 % ($a = 0{,}1$). Para seleccionar el mejor valor de este hiperparámetro se han requerido unos tiempos de ejecución de 5.78, 49.69 y 590.84 segundos respectivamente. En la Figura 4.5 podemos ver cuán sensible es la tasa de acierto de la SVM en función de los valores de `C`. Puede apreciarse cómo a partir de un cierto valor de `C` el acierto crece de manera rápida para mantenerse constante

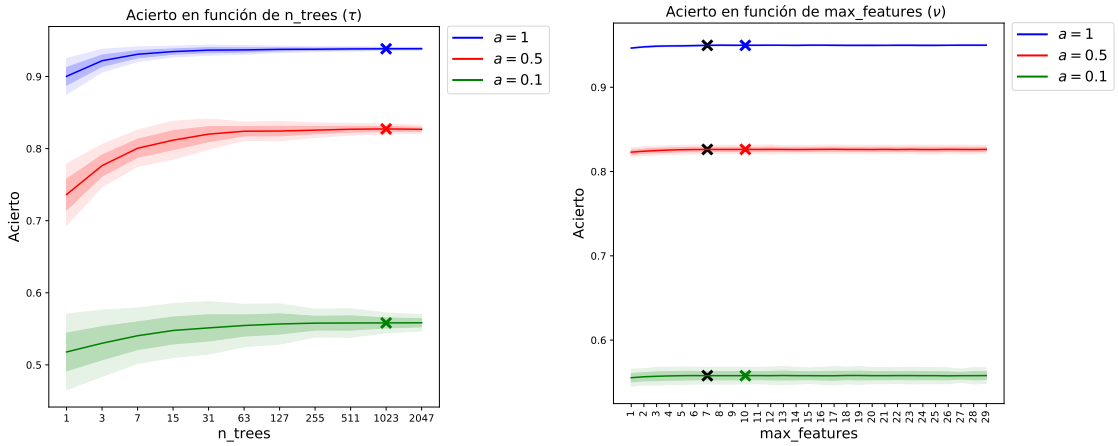

Figura 4.4: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 RF con $\nu = 10$ (izquierda). Se marcan con una cruz los valores de τ escogidos para analizar el conjunto de datos. A la derecha la tasa de acierto promedio en función de $\nu \pm$ una y dos desviaciones típicas en 50 RF con $\tau = 1023$ para cada a. Se marcan con una cruz en color el valor de ν elegido para el experimento ($\sqrt{100} = 10$). En negro se marca $\log_2(100) \sim 7$ (el logaritmo en base 2 del número total de atributos suele ser otra elección del parámetro ν).

en valores sucesivos (para los tres a).

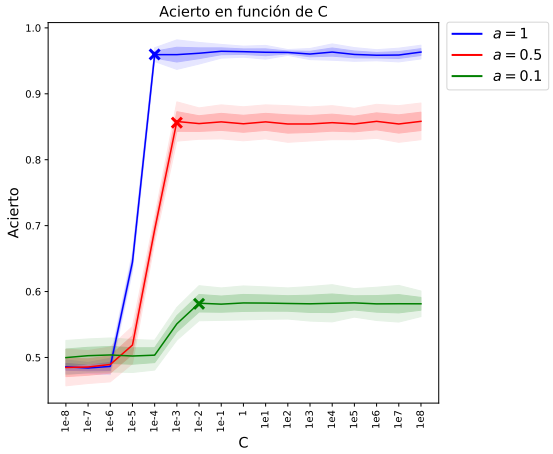

Figura 4.5: Tasa de acierto promedio en función de C $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal para cada a. Se marcan con una cruz los valores del hiperparámetro escogidos por validación cruzada.

Aplicando las SVM con núcleos lineales se obtienen tasas de acierto que mejoran en cerca de un 2 % las de los clasificadores entrenados con Random Forest (para los problemas con $a = 1$ y $a = 0{,}1$). Por el contrario, en el conjunto con $a = 0{,}5$ se consigue una tasa de acierto muy similar a la del Random Forest. En cualquiera de los casos se obtienen tasas de acierto no muy buenas comparadas con otras de las metodologías que utilizamos. Los tiempos de ejecución son muy altos si los comparamos con los requeridos por Random Forest. Esto se debe a que en el caso de RF no hemos hecho una validación cruzada para seleccionar los valores de los hiperparámetros.

Por último entrenamos una SVM con núcleo RBF. En este caso tenemos que seleccionar la mejor pareja de hiperparámetros C y γ. Los escogemos por validación cruzada. Las parejas obtenidas han sido C$= 10^{-2}$ y $\gamma = 10^{-2}$ ($a = 1$), C$= 10^{-2}$ y $\gamma = 10^{-2}$ ($a = 0{,}5$) y C$= 10^{-1}$ y $\gamma = 10^{-2}$ ($a = 0{,}1$). Para poder ver cómo varían los aciertos en función de los valores de los hiperparámetros y de a mostramos la Figura 4.6. Con estas configuraciones de los hiperparámetros obtenemos unas tasas de acierto de 95.7 % y 96.1 % para $a = 1$, 81.3 % y 81.7 % con $a = 0{,}5$ y 57.4 % y 57.8 % para

$a = 0{,}1$ en validación cruzada y test respectivamente. Se han empleado unos tiempos de ejecución de 269.03 segundos ($a = 1$), 2489.11 segundos ($a = 0{,}5$) y 29921.03 segundos ($a = 0{,}1$).

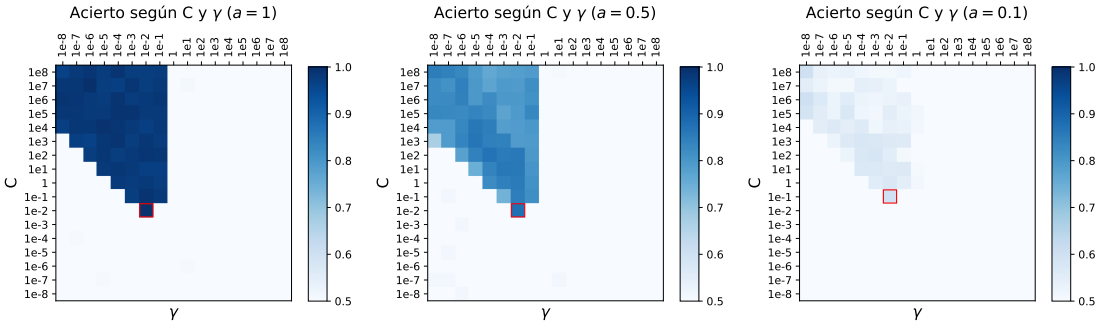

Figura 4.6: Tasa de acierto promedio en función de `C` y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

Puede verse cómo a partir de ciertos valores del hiperparámetro `C` la tasa acierto promedio se estanca. Lo contrario ocurre con γ. En este caso el acierto crece hasta que a partir de un cierto valor se desploma. Aplicando el test de Wilcoxon a los resultados con los de la SVM con núcleo lineal obtenemos que no podemos concluir que se aprecien diferencias significativas. En el problema con $a = 1$ obtenemos un $p-$valor de 0.1431. En el conjunto con $a = 0{,}5$ conseguimos uno de 0.02881 (por lo que podemos afirmar que hay una mejora significativa con respecto a los resultados obtenidos con el núcleo lineal con un nivel de significación del 95 % pero no del 99 %). Por último, en el problema con $a = 0{,}1$ se tiene un $p-$valor de 0.2799, por lo que no podemos rechazar la hipótesis nula. A pesar de obtener estos $p-$valores sí que se aprecia un claro aumento en el tiempo de ejecución con respecto a la SVM con núcleo lineal. Esto se debe a que el núcleo RBF requiere seleccionar dos hiperparámetros en lugar de uno.

Pasamos ahora a analizar el segundo método. Para generar el segundo conjunto de datos hemos empleado 10 componentes principales para los tres valores de a. Tomamos estas cantidades del número de componentes principales porque la tasa de acierto se mantiene constante para valores superiores (para los tres a), cómo puede verse en la Figura 4.7. Podríamos tomar más componentes principales para generar el conjunto de datos transformado pero no aumentaría la tasa de acierto.

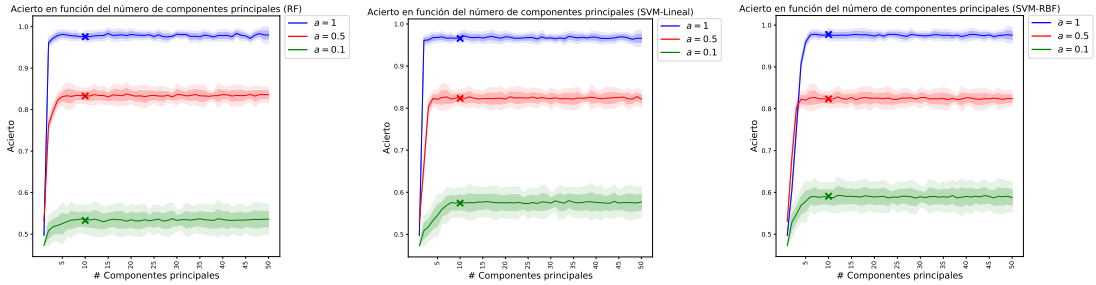

Figura 4.7: Tasa de acierto promedio ± una y dos desviaciones típicas de 50 Random Forest con $\tau = 1023$ (izquierda), SVM con núcleo lineal (centro) y SVM con núcleo RBF (derecha) en función del número de componentes principales empleadas para generar el conjunto transformado (para los tres valores de a). Se han entrenado las SVM con los valores de los hiperparámetros obtenidos por validación cruzada (para cada número de componentes principales). Se marcan con una cruz la cantidad de componentes principales escogidas (10).

Una vez tenemos los conjuntos entrenamos los Random Forest. Para hacernos una idea de cómo varían los aciertos en función de los valores del número de árboles del RF (τ) podemos ver la Figura 4.8. Como puede verse tomar como del hiperparámetro `n_estimators` 1023 sigue siendo razonable. Además se puede apreciar cómo la tasa de acierto tiende a un valor asintótico y deja de crecer. Con estas configuraciones de los hiperparámetros obtenemos unas tasas de acierto en test

de 95.5 % ($a = 1$), 81.5 % ($a = 0{,}5$) y 54.3 ($a = 0{,}1$). Se han requerido unos tiempos de ejecución de 0.78 segundos ($a = 1$), 0.81 segundos ($a = 0{,}5$) y 0.79 segundos ($a = 0{,}1$).

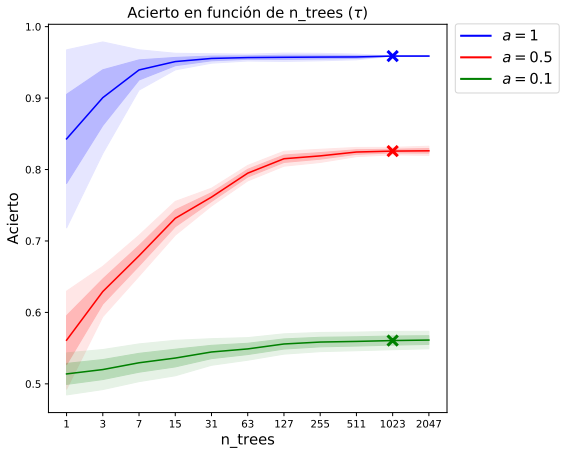

Figura 4.8: Tasa de acierto promedio en función de τ ± una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 10$ (para cada a). Se marcan con una cruz los valores del hiperparámetro escogidos.

Tras entrenar los RF con las componentes principales del conjunto original obtenemos una mejoría con respecto a la metodología anterior. En los problemas con $a = 1$ y $a = 0{,}5$ el acierto crece en cerca del 1.5 %, mientras que en el conjunto con $a = 0{,}1$ este aumento es cercano al 1 %. Además los tiempos de ejecución se han reducido levemente. Esto se debe a que la dimensión de los datos se ha reducido drásticamente.

Pasamos ahora a entrenar una SVM con núcleo lineal. Seleccionamos el valor del hiperparámetro `C` por validación cruzada. Para cada valor de a tenemos que el valor del hiperparámetro escogido es `C` $= 10^{-3}$ para los tres valores de a. Las tasas de acierto en validación cruzada y test para cada a son 96.3 % y 96.8 % en 3.05 segundos ($a = 1$), 81.9 % y 82.4 % en 30.27 segundos ($a = 0{,}5$) y 57.2 % y 57.6 % en 371.12 segundos ($a = 0{,}1$). En la Figura 4.9 podemos ver cómo varía la tasa de acierto en función del valor del hiperparámetro `C`. El acierto se mantiene muy bajo para valores pequeños de `C` y crece rápidamente a partir de un valor umbral. Después sigue manteniéndose constante. Este patrón se repite en los tres valores de a.

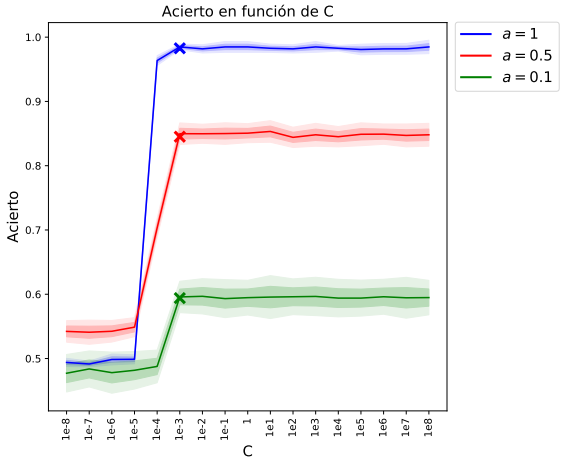

Figura 4.9: Tasa de acierto promedio en función de `C` ± una y dos desviaciones típicas en 50 SVM con núcleo lineal para cada a. Se marcan con una cruz los valores del hiperparámetro escogidos por validación cruzada.

Con este clasificador se tienen unos resultados mejores que al utilizar la metodología anterior. En los tres conjuntos de datos obtenemos resultados claramente mejores que cuando aplicamos

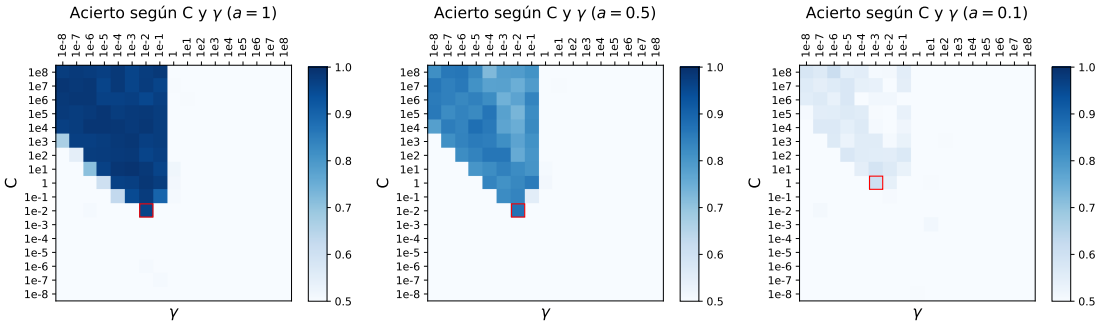

Figura 4.10: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo valores de los hiperparámetros escogidos por validación cruzada.

la SVM lineal al conjunto original aunque no son lo suficientemente mejores cómo para haya una clara evidencia estadística que lo soporte. No obstante los tiempos de ejecución empleados para seleccionar las configuraciones de los hiperparámetros se han reducido notablemente. Esto se debe a que se ha disminuido el número de atributos del conjunto.

Por último entrenamos una SVM con núcleo RBF. Seleccionamos los valores de los hiperparámetros C y γ por validación cruzada. Para cada valor de a tenemos que la pareja de hiperparámetros escogida es $C = 10^{-2}$ y $\gamma = 10^{-2}$ para $a = 1$ y $a = 0{,}5$ y $C = 10^{-1}$ y $\gamma = 10^{-2}$ en el caso de $a = 0{,}1$. Las tasas de acierto en validación cruzada y test para cada a son 97.2 % y 97.6 % en 110.52 segundos ($a = 1$), 82.0 % y 82.5 % en 918.21 segundos ($a = 0{,}5$) y 58.7 % y 59.0 % en 10267.88 segundos ($a = 0{,}1$). En la Figura 4.10 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Para valores grandes de γ el error es muy alto. Lo mismo ocurre para valores pequeños de C. Sin embargo, cuando C es grande y γ pequeño la tasa de acierto es alta (para los tres valores de a). Las mejores tasas de acierto se obtienen en un valor intermedio. Este valor coincide con las configuraciones óptimas de los hiperparámetros halladas por validación cruzada.

A pesar de obtener mejores tasas de acierto que al entrenar los mismos clasificadores con el conjunto original, aplicando el test de Wilcoxon no obtenemos evidencia estadística suficiente como para afirmar que existe una diferencia significativa. No obstante sí se puede apreciar cómo los tiempos de ejecución se han reducido en gran medida. Esto se debe a que hemos reducido notablemente el número de atributos.

A continuación utilizamos el tercer conjunto de datos. Para ello hemos proyectado las funciones originales en la base de Fourier con 4, 4 y 13 elementos para los valores de $a = 1$, $a = 0{,}5$ y $a = 0{,}1$ respectivamente. Seleccionamos estas cantidades del parámetro n_F por validación cruzada utilizando RF (en las Figuras 5.1 y 5.2 de la Sección 5.2.1 del apéndice pueden verse las gráficas de las elecciones por validación cruzada para los clasificadores SVM con núcleos lineal y RBF). En la Figura 4.11 puede verse la tasa de acierto en validación y en test. Además se aprecia cómo proyectar las funciones en bases de Fourier con más elementos hace que disminuya la tasa de acierto (se añade ruido). Este acierto crece según aumenta el número de elementos en la base de Fourier hasta un cierto valor umbral (que es distinto para cada a). Después decrece lentamente según aumenta el parámetro n_F.

Para hacernos una idea de que aspecto tienen las funciones proyectadas en la base de Fourier, mostramos cinco funciones de cada clase para el conjunto con $a = 1$. En la Figura 4.12 podemos ver cómo las funciones se han suavizado y parecen separarse mejor en los primeros instantes de tiempo. Esto nos hace pensar que, con este conjunto de datos, se obtendrán mejores tasas de acierto que con el conjunto original.

Entrenamos los Random Forest. Para ello tomamos los valores del hiperparámetro `n_estimators` 1023. En la Figura 4.13 podemos ver cómo esta elección sigue siendo razonable para estos conjuntos de datos. Además aquí podemos observar cómo varían los aciertos en función de los valores de τ. Las tasas de acierto (para los tres valores de a) crecen según lo hace τ hasta estancarse en un valor asintótico. Obtenemos unas tasas de acierto de 95.9 % ($a = 1$), 81.9 % ($a = 0{,}5$) y 54.3 ($a = 0{,}1$) en test en un total de 0.75, 0.79 y 0.80 segundos para cada a.

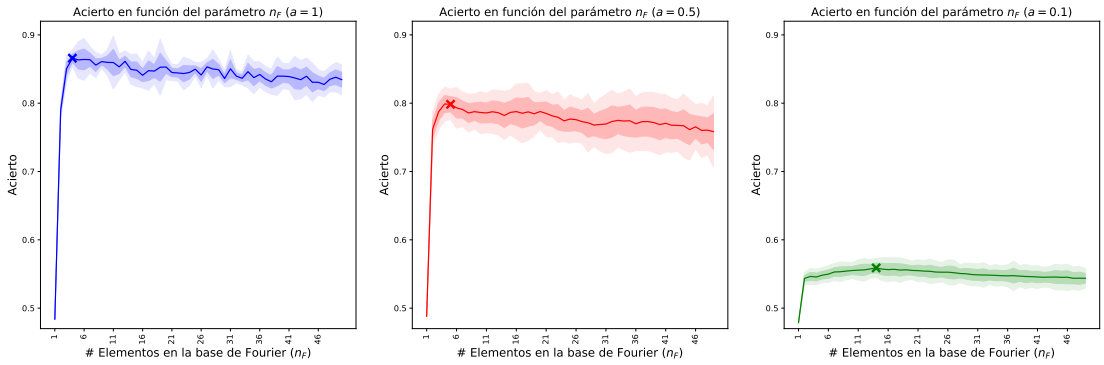

Figura 4.11: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de elementos de la base de Fourier (n_F) y para los tres valores de a. Se marcan con una cruz los valores de n_F escogidos por validación cruzada.

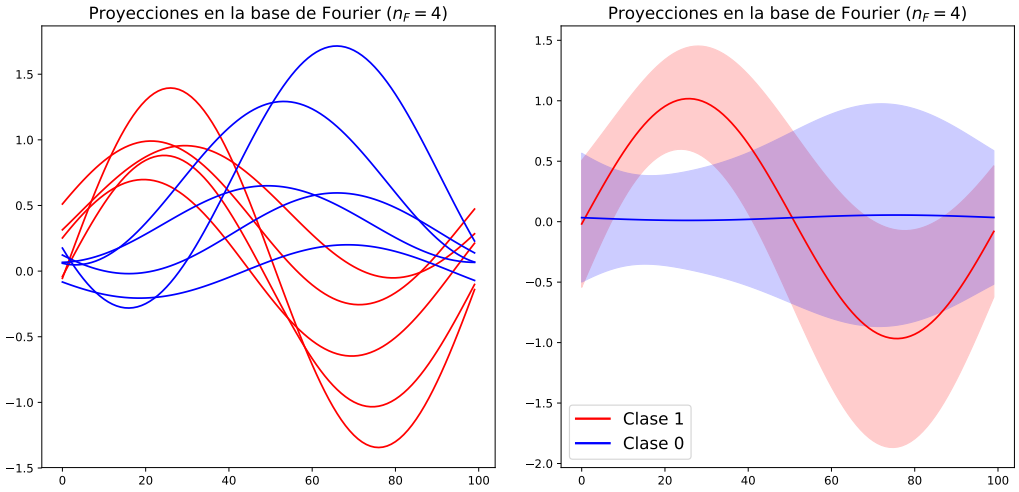

Figura 4.12: A la izquierda, las proyecciones de 5 funciones de cada clase del conjunto con $a = 1$ en la base de Fourier con 4 elementos ($n_F = 4$). A la derecha las medias de cada clase ± una desviación típica de todo el conjunto (con $a = 1$) proyectado en esta base.

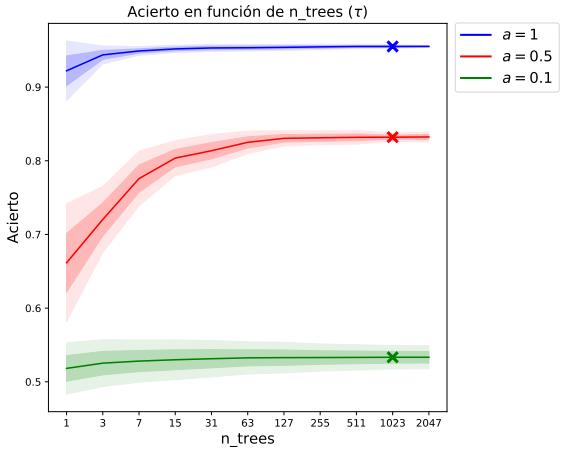

Figura 4.13: Tasa de acierto promedio en función de τ ± una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 10$ (para cada a). Se marcan con una cruz los valores del hiperparámetro seleccionados.

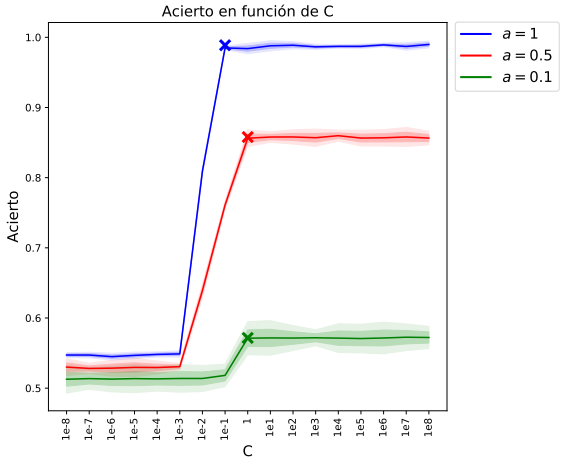

Figura 4.14: Tasa de acierto promedio en función de C ± una y dos desviaciones típicas en 50 SVM con núcleo lineal para cada a. Se marcan con una cruz los valores del hiperparámetro escogidos por validación cruzada.

En los tres conjuntos de datos (para los tres valores de a) podemos observar una leve mejoría con respecto a los aciertos conseguidos usando la metodología de las componentes principales y un gran aumento con respecto a los clasificadores entrenados con el conjunto original (cerca del 2 % en los problemas con $a = 1$ y $a = 0{,}5$ y del 1 % en el correspondiente con $a = 0{,}1$). Además, a pesar de trabajar con 9, 9 y 27 atributos para los valores de a 1, 0,5 y 0,1 respectivamente, podemos representar las funciones proyectadas para hacernos una idea de su aspecto (véase la Figura 4.12). Esto es una clara ventaja con respecto a la metodología de las componentes principales (en este caso sólo podemos visualizar las transformaciones cuando se utilizan 2 o 3 componentes principales).

Ahora entrenamos una SVM con núcleo lineal. Escogemos el valor del hiperparámetro C por validación cruzada. Para cada valor de a tenemos que el valor del hiperparámetro seleccionado es $C = 10^{-1}$ en el conjunto con $a = 1$ y $C = 1$ en los problemas con $a = 0{,}5$ y $a = 0{,}1$. Las tasas de acierto en validación cruzada y test para cada a son 96.4 % y 96.7 % en 3.32 segundos ($a = 1$), 81.9 % y 82.3 % en 18.23 segundos ($a = 0{,}5$) y 57.3 % y 57.8 % en 207.41 segundos ($a = 0{,}1$). En la Figura 4.14 podemos ver cómo varía la tasa de acierto en función de los diferentes valores del hiperparámetro C de la malla. Las tasas de acierto son bajas para valores pequeños de C. Llegado un cierto valor del hiperparámetro crecen muy rápidamente para mantenerse constantes para valores superiores.

Con las configuraciones de los hiperparámetros escogidos por validación cruzada obtenemos resultados similares a los de los clasificadores entrenados usando la metodología de las componentes principales. Se obtienen resultados visiblemente mejores que cuando se utiliza el conjunto de datos original pero no se consigue evidencia estadística suficiente en el test de Wilcoxon.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros C y γ usamos una validación cruzada. Para cada valor de a tenemos que la pareja de hiperparámetros escogida es $C = 10^{-2}$ y $\gamma = 10$ para $a = 1$ y $a = 0{,}5$ y $C = 10^{-1}$ y $\gamma = 1$ para el problema con $a = 0{,}1$. En la Figura 4.15 podemos ver cómo varían las tasas de acierto en función cada pareja de hiperparámetros C y γ. Los aciertos se mantienen altos para valores grandes de C y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. Las tasas de acierto de validación cruzada y test para cada a son 97.4 % y 97.0 % en 132.15 segundos ($a = 1$), 82.1 % y 82.4 % en 771.28 segundos ($a = 0{,}5$) y 59.1 % y 59.2 % en 9268.14 segundos ($a = 0{,}1$).

Los resultados obtenidos son muy similares a los que se tienen cuando utilizamos las SVM con núcleo lineal. Las tasas de acierto son muy parecidas a las que se tienen cuando entrenamos los clasificadores aplicando la metodología de las componentes principales. Tampoco se obtiene evidencia estadística suficiente para poder afirmar, aplicando el test de Wilcoxon, que hay una clara mejoría con respecto a la metodología original.

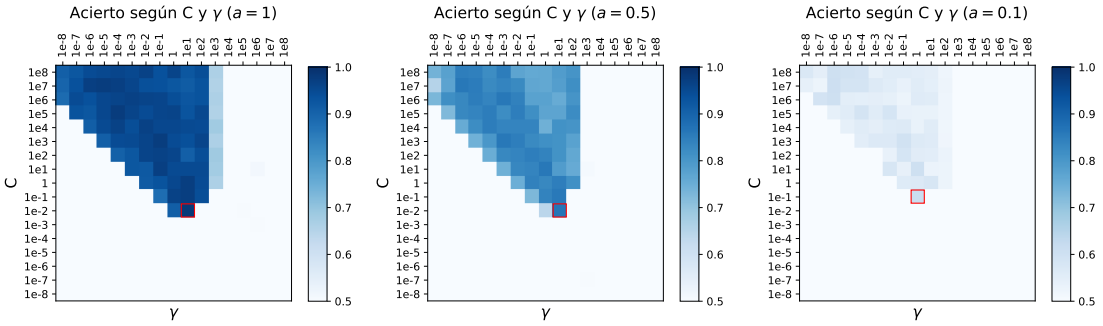

Figura 4.15: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

A continuación juntamos los dos conjuntos de datos anteriores (las componentes principales y los coeficientes en las bases de Fourier) para cada a. Con cada conjunto agrupado entrenamos un Random Forest. Utilizamos los mismos valores del hiperparámetro τ. En la Figura 4.16 podemos observar cómo varían los aciertos en función de τ. También podemos ver cómo tomar $\tau = 1023$ sigue siendo una elección razonable. Puede apreciarse el mismo fenómeno que con los Random Forest anteriores. El acierto crece con τ hasta un valor asintótico. Esto ocurre para los tres valores de a. Obtenemos unas tasas de acierto en test del 96.2 % ($a = 1$), 82.0 % ($a = 0{,}5$) y 54.9 % ($a = 0{,}1$) respectivamente. Se ha necesitado un total de 0.83 segundos para $a = 1$ y $a = 0{,}5$ y 0.84 segundos para $a = 0{,}1$.

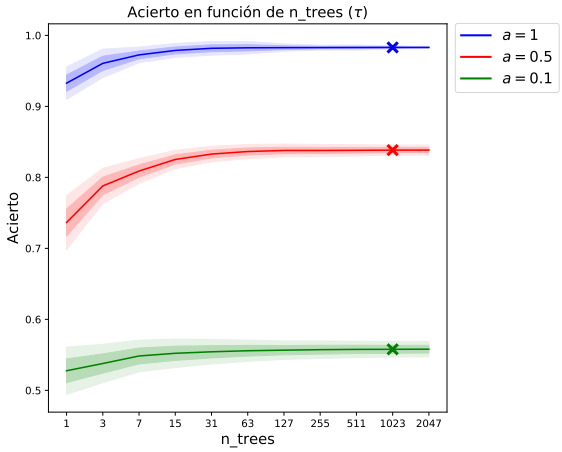

Figura 4.16: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 10$ (para cada a). Se marcan con una cruz los valores del hiperparámetro seleccionados.

Es con estos conjuntos de datos con los que se consiguen la mejores tasas de acierto (para los tres valores de a). Las tasas de acierto aumentan en cerca de un 3 % en los conjuntos con $a = 1$ y $a = 0{,}5$ con respecto a los resultados de la metodología original. En el caso de $a = 0{,}1$ este aumento es de aproximadamente 1,5 %. Esto nos hace pensar que, cuando apliquemos las SVM obtendremos resultados significativamente mejores usando el test de Wilcoxon.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro C por validación cruzada. Para cada valor de a tenemos que el valor del hiperparámetro escogido es $C = 10^{-4}$ para los conjuntos con $a = 1$ y $a = 0{,}5$ y $C = 10^{-3}$ en el problema con $a = 0{,}1$. En la Figura 4.17 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro C. Se tiene un error alto para valores pequeños de C. Este decrece rápidamente hasta mantenerse constante en valores de Csuperiores. Las tasas de acierto en validación cruzada y test para cada a son 96.9 % y 97.2 % en 4.24 segundos ($a = 1$), 82.5 % y 82.6 % en 39.97 segundos ($a = 0{,}5$) y 60.0 % y 60.2 % en 475.45 segundos ($a = 0{,}1$).

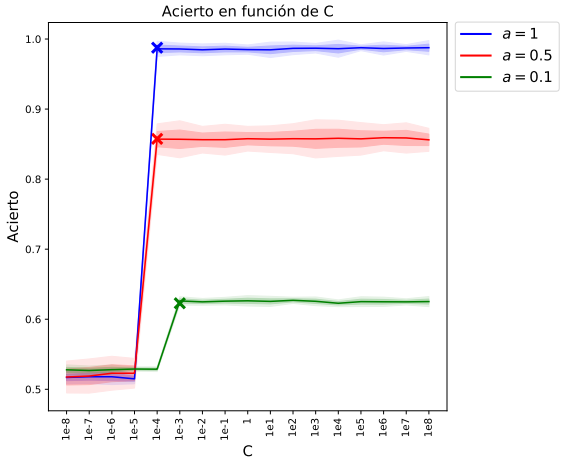

Figura 4.17: Tasa de acierto promedio en función de C ± una y dos desviaciones típicas en 50 SVM con núcleo lineal para cada a. Se marcan con una cruz los valores del hiperparámetro escogidos por validación cruzada.

Es aplicando esta metodología cuando obtenemos las mejores tasas de acierto (para los tres valores de a) de entre todas las SVM con núcleo lineal. Además obtenemos resultados significativamente mejores para cualquier nivel de significación habitual si los comparamos con los resultados de entrenar los clasificadores con los conjuntos originales. Aplicando el test de Wilcoxon obtenemos los $p-$valores valen 0,0009 para $a = 1$, 0,0001 para $a = 0{,}5$ y 0,0014 para $a = 0{,}1$. Además los tiempos requeridos para seleccionar las configuraciones de los hiperparámetros es bastante menor.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros C y γ usamos una validación cruzada. Para cada valor de a tenemos que la pareja de hiperparámetros escogida es $C = 10^{-2}$ y $\gamma = 10^{-2}$ en el problema con $a = 1$ y $C = 10^{-1}$ y $\gamma = 10^{-2}$ en los problemas con $a = 0{,}5$ y $a = 0{,}1$. En la Figura 4.18 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Se puede apreciar el mismo comportamiento que en las SVM con núcleo RBF anteriores. Los aciertos son altos para valores grandes de C y pequeños de γ. en cualquier otro caso los aciertos son muy bajos. Hay un valor intermedio de los hiperparámetros donde el error es mínimo. Este valor intermedio coincide con las configuraciones obtenidas mediante validación cruzada. Las tasas de acierto en validación cruzada y test para cada a son 98.4 % y 98.8 % en 208.47 segundos ($a = 1$), 82.9 % y 83.2 % en 1890.51 segundos ($a = 0{,}5$) y 60.2 % y 60.6 % en 21655.71 segundos ($a = 0{,}1$).

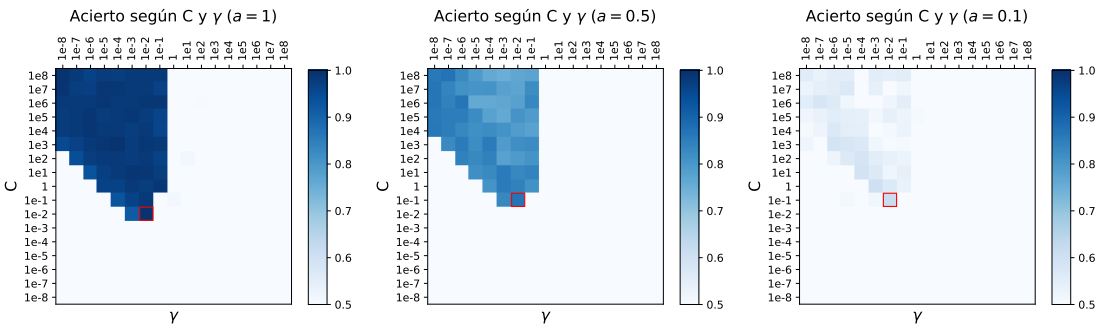

Figura 4.18: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo valores de los hiperparámetros escogidos por validación cruzada.

Al igual que antes, usando este conjunto de datos, obtenemos las mejores tasas de acierto del experimento. Los resultados son significativamente mejores que cuando se aplica el conjunto de datos original para entrenar los clasificadores. Aplicando el test de Wilcoxon obtenemos que los $p-$valores valen 0,0008 en el caso de los resultados con $a = 1$, 0,0016 en el de $a = 0{,}5$ y 0,0012 en el de $a = 0{,}1$. Por lo tanto hay evidencia estadística suficiente para afirmar que las tasas de acierto

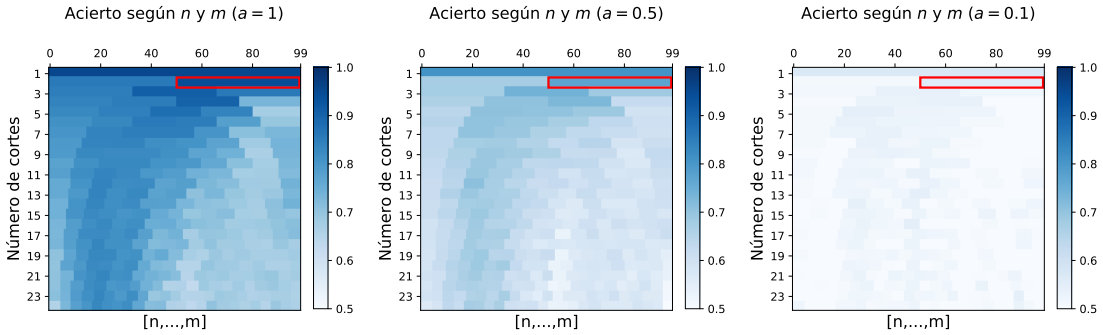

Figura 4.19: Tasas de acierto promedio de 50 Random Forest para los conjuntos troceados por los instantes $[n, \ldots, m]$ para cada a.

de los clasificadores entrenados usando la metodología de juntar las componentes principales y los coeficientes de Fourier son mejores que los obtenidos utilizando el conjunto original.

Pasamos a utilizar la técnica del troceado de las funciones. Troceamos las funciones originales en varias subfunciones. Primero, a partir de cada función original, generamos dos nuevas subfunciones. La primera consistirá en los instantes $[0, \ldots, 49]$ de las originales y la segunda en los instantes $[50, \ldots, 99]$. Troceamos las funciones originales de esta manera y entrenamos 50 Random Forest con las nuevas subfunciones. Las tasas de acierto promedio obtenidas a partir de este troceado pueden verse en la segunda fila de las gráficas de la Figura 4.19. Después hacemos lo mismo cortando las funciones originales por los instantes $[0, \ldots, 32]$, $[33, \ldots, 65]$ y $[66, \ldots, 99]$. Con las funciones cortadas por estos instantes generamos 50 RF y mostramos las tasas de acierto promedio en la tercera fila de las gráficas de la Figura 4.19. Repetimos esto hasta que, a partir de la función original, podemos generar 25 nuevas subfunciones.

En la Figura 4.19 mostramos los aciertos de 50 RF que han sido entrenados con las funciones cortadas por los instantes $[n \ldots, m]$. Puede verse cómo, según se toman intervalos más pequeños el acierto disminuye. Además el acierto es más grande para valores pequeños de n y m (el principio de las funciones originales parece dar más información que el final). Sin embargo la mejor tasa de acierto se tiene cuando se trocean las funciones originales por los instantes $[50, \ldots, 99]$. Estos son los valores de n y m que utilizamos para generar las subfunciones del conjunto de datos troceado (para todos los a).

Una vez hemos troceado las funciones entrenamos un Random Forest. Utilizamos los mismos valores del hiperparámetro τ que en los casos anteriores. En la Figura 4.20 podemos observar cómo varía la tasa de acierto en función de los valores de τ de la malla. Además se puede apreciar cómo la elección de $\tau = 1023$ sigue siendo razonable. Las tasas de acierto (para los tres valores de a) crecen según lo hace τ hasta estancarse en un valor asintótico. Se obtienen unas tasas de acierto del 92.3 % ($a = 1$), 78.1 % ($a = 0{,}5$) y del 53.1 % ($a = 0{,}1$) en test respectivamente. Se ha empleado un total de 0.81 segundos, 0.82 segundos y 0.80 segundos para cada a.

Con esta metodología de trocear las funciones originales obtenemos los peores resultados de todo el experimento. Se logran tasas de acierto menores que entrenando los Random Forest con las funciones originales. No obstante el tiempo de ejecución empleado en entrenar los clasificadores se reduce levemente. Esto se debe a que se disminuye a la mitad el número de atributos de los datos.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro `C` por validación cruzada. Para cada valor de a tenemos que el valor del hiperparámetro escogido es $\texttt{C} = 10^{-3}$ en el primer problema ($a = 1$) y $\texttt{C} = 10^{-2}$ en los dos restantes ($a = 0{,}5$ y $a = 0{,}1$). En la Figura 4.21 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro `C`. Las tasas de acierto son bajas para valores pequeños de `C`. Llegado un cierto valor del hiperparámetro crecen muy rápidamente para mantenerse constantes para valores superiores. Las tasas de acierto de validación cruzada y test para cada a son 94.4 % y 94.8 % en 5.15 segundos ($a = 1$), 77.2 % y 77.7 % en 49.12 segundos ($a = 0{,}5$) y 55.1 % y 55.5 % en 601.73 segundos ($a = 0{,}1$).

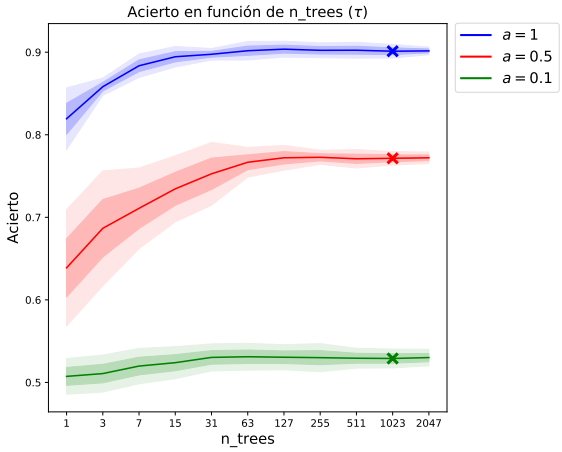

Figura 4.20: Tasa de acierto promedio en función de τ $\pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 10$ (para cada a). Se marcan con una cruz los valores del hiperparámetro seleccionados.

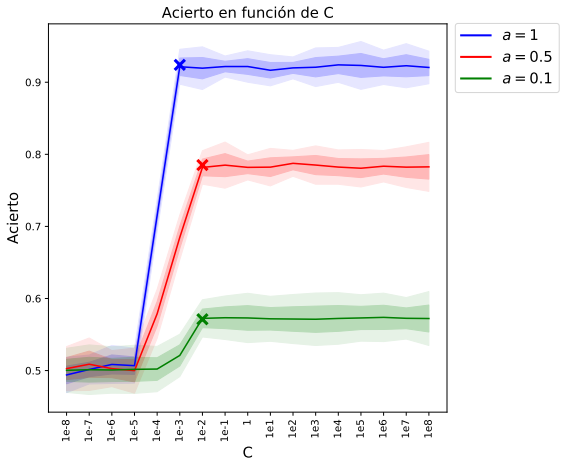

Figura 4.21: Tasa de acierto promedio en función de C $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal para cada a. Se marcan con una cruz los valores del hiperparámetro escogidos por validación cruzada.

Los aciertos de las SVM entrenados a partir de los datos originales troceados también obtienen las peores tasas de acierto de entre todas las metodologías utilizadas. Si comparamos la tasa de acierto del problema con $a = 0{,}5$ con el acierto de la SVM con núcleo lineal entrenado a partir del conjunto original, obtenemos evidencia estadística suficiente (en el test de Wilcoxon) para afirmar que, a cualquier nivel de significación habitual, se obtienen peores resultados. El $p-$valor en este caso vale 0,0075.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros C y γ usamos una validación cruzada. Para cada valor de a tenemos que la pareja de hiperparámetros escogida es $\mathtt{C} = 10^{-1}$ y $\gamma = 10^{-1}$ en los dos primeros problemas ($a = 1$ y $a = 0{,}5$) y $\mathtt{C} = 1$ y $\gamma = 1$ en el último ($a = 0{,}1$). En la Figura 4.22 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Los aciertos se mantienen altos para valores grandes de C y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. Las tasas de acierto de validación cruzada y test para cada a son 95.6 % y 96.0 % en 241.24 segundos ($a = 1$), 79.1 % y 79.4 % en 2785.12 segundos ($a = 0{,}5$) y 57.2 % y 57.6 % en 25297.36 segundos ($a = 0{,}1$).

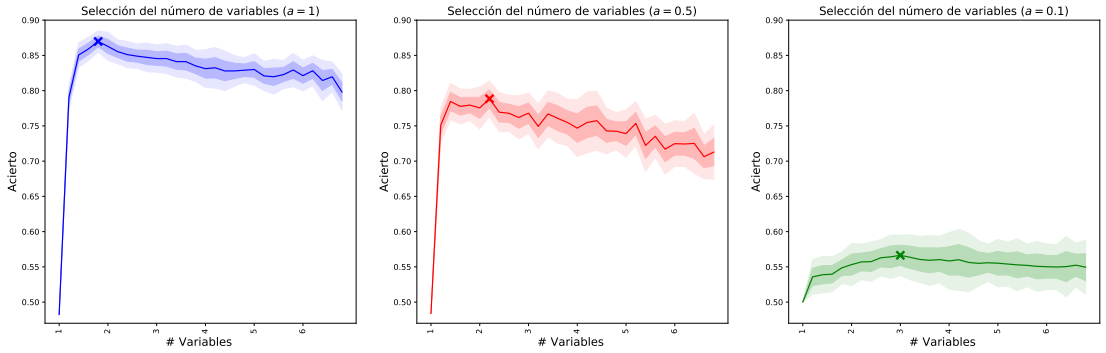

Figura 4.23: Tasas de acierto promedio en test y validación cruzada ± una y dos desviaciones típicas en función del número de variables seleccionadas (para cada valor de a). Se marcan con una cruz el número de variables seleccionadas por validación cruzada.

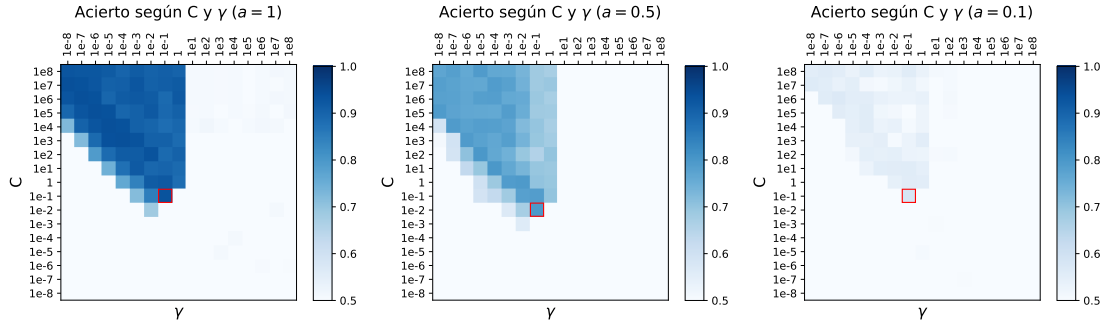

Figura 4.22: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

Las tasas de acierto de estos clasificadores son peores que los que se obtienen al utilizar el conjunto de datos original. A pesar de no haber evidencia estadística suficiente que afirmarlo con un nivel de significación del 99 %, todo parece indicar que al trocear las funciones estamos desechando información relevante.

Por último reducimos la dimensión de los datos mediante una selección de variables de los instantes de las funciones. Seleccionamos la cantidad de instantes (variables) mediante una validación cruzada. De esta manera obtenemos que vamos a seleccionar 5 variables (instantes temporales) en el problema con $a = 1$, 7 en el problema con $a = 0{,}5$ y 11 en el problema con $a = 0{,}1$. En la Figura 4.23 podemos ver las tasas de acierto de validación cruzada y test para cada conjunto de datos usando Random Forest (en las Figuras 5.3 y 5.4 de la Sección 5.2.1 del apéndice pueden verse las gráficas de las elecciones por validación cruzada para los clasificadores SVM con núcleos lineal y RBF).

Entrenamos un Random Forest. Utilizamos los mismos valores del hiperparámetro τ que en los casos anteriores. En la Figura 4.24 podemos observar cómo varía la tasa de acierto en función de los valores de τ de la malla. Además se puede apreciar cómo las elecciones de $\tau = 1023$ siguen siendo adecuadas. Las tasas de acierto (para los tres valores de a) crecen según lo hace τ hasta estancarse en un valor asintótico. De esta forma se obtienen unas tasas de acierto en test del 94.8 % ($a = 1$), 79.3 % ($a = 0{,}5$) y 53.4 % ($a = 0{,}1$). Se ha requerido un total de 0.72, 0.75 y 0.76 segundos para cada a.

Utilizando esta metodología para entrenar los Random Forest obtenemos unas tasas de acierto ligeramente superiores que cuando usamos todo el conjunto original (para los tres valores de a). Aunque no obtenemos evidencia estadística suficiente para concluir que los resultados son mejores a un nivel de significación elevado, sí podemos observar que los tiempos de ejecución han disminuido bastante. Además podemos representar fácilmente las transformaciones de las funciones (véase la Figura 4.23).

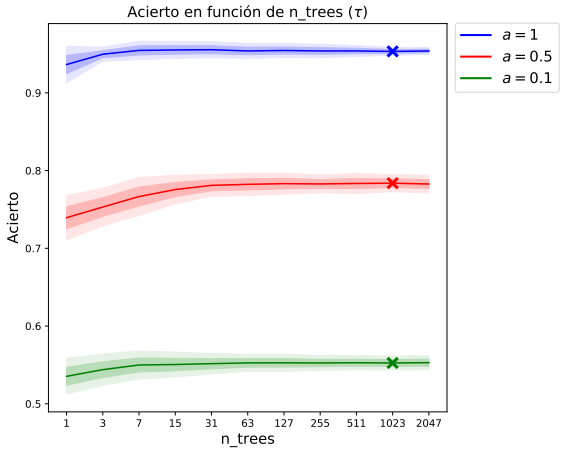

Figura 4.24: Tasa de acierto promedio en función de τ $\pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 10$ (para cada a). Se marcan con una cruz los valores del hiperparámetro seleccionados.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro C por validación cruzada. Para cada valor de a tenemos que el valor del hiperparámetro escogido es $\texttt{C} = 10^{-2}$ en los problemas con $a = 1$ y $a = 0{,}5$ y $\texttt{C} = 10^{-1}$ en el restante ($a = 0{,}1$). En la Figura 4.25 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro C. Las tasas de acierto son bajas para valores pequeños de C. Llegado un cierto valor del hiperparámetro crecen muy rápidamente para mantenerse constantes para valores superiores. Las tasas de acierto de validación cruzada y test para cada a son 94.9 % y 95.3 % en 1.07 segundos ($a = 1$), 79.2 % y 79.6 % en 10.05 segundos ($a = 0{,}5$) y 55.4 % y 55.7 % en 109.21 segundos ($a = 0{,}1$).

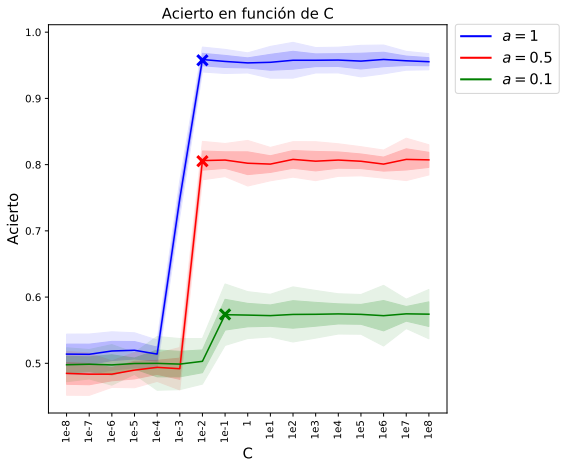

Figura 4.25: Tasa de acierto promedio en función de C $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal para cada a. Se marcan con una cruz los valores del hiperparámetro escogidos por validación cruzada.

Los resultados que se obtienen al utilizar las SVM con núcleo lineal son similares a los que se consiguen con los Random Forest. Las tasas de acierto son del mismo orden de magnitud pero los tiempos de ejecución empleados en hallar las mejores configuraciones de los hiperparámetros son mucho menores.

Por último entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros C y γ usamos una validación cruzada. Para cada valor de a tenemos que la pareja de hiperparámetros escogida es $\texttt{C} = 10^{-1}$ y $\gamma = 1$ en los tres conjuntos ($a = 1$, $a = 0{,}5$ y $a = 0{,}1$). En la Figura 4.26 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Los aciertos se mantienen altos para valores grandes de C y pequeños de

γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. Las tasas de acierto de validación cruzada y test para cada a son 96.1 % y 96.4 % en 68.55 segundos ($a = 1$), 81.4 % y 81.8 % en 628.54 segundos ($a = 0{,}5$) y 57.8 % y 58.1 % en 7509.86 segundos ($a = 0{,}1$).

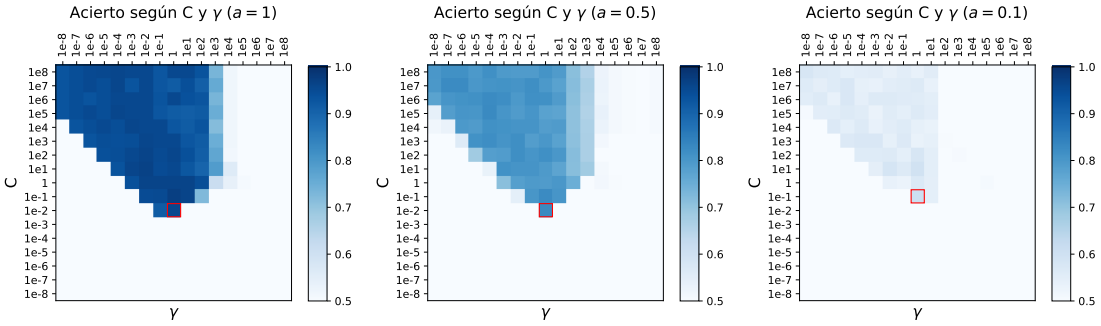

Figura 4.26: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

En este caso obtenemos resultados ligeramente superiores que cuando utilizamos el conjunto de datos original para entrenar los clasificadores. No se obtiene evidencia estadística suficiente para concluir esta afirmación a un nivel de significación del 99 % pero sí se puede apreciar una clara mejoría con respecto a las SVM con núcleo lineal. En el caso de los problemas con $a = 0{,}5$ y $a = 0{,}1$, el test de Wilcoxon nos permite concluir que, al utilizar esta metodología, es mejor entrenar una SVM con núcleo RBF frente a una con núcleo lineal (los $p-$valores de los contrastes valen 0,0071 y 0,0064 para $a = 0{,}5$ y $a = 0{,}1$ respectivamente).

A modo de resumen, mostramos a continuación una tabla con los resultados del experimento con estos conjuntos de datos.

Algoritmo	Método	$\mu_{CV}(\sigma_{CV})$	$\mu_{test}(\sigma_{test})$	t
RF	Original	X	0.929	0.98
RF	10 componentes principales	X	0.955	0.78
RF	Fourier ($n_F = 4$)	X	0.959	0.75
RF	Agrupados	X	**0.962**	0.83
RF	Troceados	X	0.923	0.81
RF	5 coordenadas	X	0.948	0.72
SVM-Lineal	Original	0.946 (0.015)	0.950 (0.013)	5.78
SVM-Lineal	10 componentes principales	0.963 (0.011)	0.968 (0.012)	3.05
SVM-Lineal	Fourier ($n_F = 4$)	0.964 (0.011)	0.967 (0.010)	3.32
SVM-Lineal	Agrupados	0.969 (0.012)	**0.972** (0,012)***	4.24
SVM-Lineal	Troceados	0.945 (0.013)	0.948 (0.011)	5.15
SVM-Lineal	5 coordenadas	0.949 (0.013)	0.953 (0.012)	1.07
SVM-RBF	Original	0.957 (0.012)	0.961 (0.013)	269.03
SVM-RBF	10 componentes principales	0.972 (0.013)	0.976 (0.013)	110.52
SVM-RBF	Fourier ($n_F = 3$)	0.974 (0.012)	0.970 (0.012)	132.15
SVM-RBF	Agrupados	0.985 (0.012)	**0.988** (0,011)***	208.47
SVM-RBF	Troceados	0.957 (0.014)	0.960 (0.012)	241.24
SVM-RBF	5 coordenadas	0.961 (0.011)	0.964 (0.010)	68.55

Cuadro 4.1: Tasas de acierto promedio (validación cruzada y test) y tiempos de ejecución (en segundos) de cada algoritmo en función de la metodología utilizada para el conjunto de datos sintético con $a = 1$. Se marcan un *, ** y *** los resultados que son significativamente mejores en el test de Wilcoxon comparados con la metodología original a un nivel de significación del 90 %, 95 % y 99 % respectivamente. En negrita se marca el mejor resultado para cada familia de algoritmos.

Algoritmo	Método	$\mu_{CV}(\sigma_{CV})$	$\mu_{test}(\sigma_{test})$	t
RF	Original	X	0.791	0.93
RF	10 componentes principales	X	0.816	0.81
RF	Fourier ($n_F = 4$)	X	0.819	0.79
RF	Agrupados	X	**0.820**	0.83
RF	Troceados	X	0.782	0.82
RF	7 coordenadas	X	0.794	0.75
SVM-Lineal	Original	0.787 (0.017)	0.791 (0.016)	49.69
SVM-Lineal	10 componentes principales	0.819 (0.016)	0.824 (0.016)	30.27
SVM-Lineal	Fourier ($n_F = 5$)	0.819 (0.018)	0.823 (0.018)	28.23
SVM-Lineal	Agrupados	0.825 (0.019)	**0.826** (0,019)***	39.97
SVM-Lineal	Troceados	0.772 (0.017)	0.777 (0.017)	49.12
SVM-Lineal	7 coordenadas	0.792 (0.017)	0.796 (0.016)	10.05
SVM-RBF	Original	0.813 (0.018)	0.818 (0.018)	2489.11
SVM-RBF	10 componentes principales	0.820 (0.017)	0.825 (0.018)	918.31
SVM-RBF	Fourier ($n_F = 5$)	0.821 (0.018)	0.824 (0.017)	971.28
SVM-RBF	Agrupados	0.829 (0.017)	**0.832** (0,018)***	1890.51
SVM-RBF	Troceados	0.791 (0.018)	0.794 (0.018)	2185.12
SVM-RBF	7 coordenadas	0.814 (0.019)	0.818 (0.017)	628.54

Cuadro 4.2: Tasas de acierto promedio (validación cruzada y test) y tiempos de ejecución (en segundos) de cada algoritmo en función de la metodología utilizada para el conjunto de datos sintético con $a = 0{,}5$. Se marcan un *, ** y *** los resultados que son significativamente mejores en el test de Wilcoxon comparados con la metodología original a un nivel de significación del 90 %, 95 % y 99 % respectivamente. En negrita se marca el mejor resultado para cada familia de algoritmos.

Algoritmo	Método	$\mu_{CV}(\sigma_{CV})$	$\mu_{test}(\sigma_{test})$	t
RF	Original	X	0.533	0.97
RF	10 componentes principales	X	0.543	0.79
RF	Fourier ($n_F = 13$)	X	0.543	0.80
RF	Agrupados	X	**0.549**	0.84
RF	Troceados	X	0.531	0.80
RF	11 coordenadas	X	0.534	0.76
SVM-Lineal	Original	0.553 (0.024)	0.557 (0.026)	590.84
SVM-Lineal	10 componentes principales	0.572 (0.030)	0.576 (0.019)	371.12
SVM-Lineal	Fourier ($n_F = 13$)	0.573 (0.027)	0.578 (0.025)	407.41
SVM-Lineal	Agrupados	0.600 (0.024)	**0.602** (0,028)***	475.45
SVM-Lineal	Troceados	0.551 (0.024)	0.555 (0.025)	601.73
SVM-Lineal	11 coordenadas	0.554 (0.021)	0.557 (0.026)	109.21
SVM-RBF	Original	0.575 (0.027)	0.578 (0.031)	29921.03
SVM-RBF	10 componentes principales	0.588 (0.019)	0.590 (0.024)	10267.88
SVM-RBF	Fourier ($n_F = 13$)	0.591 (0.024)	0.595 (0.027)	11268.14
SVM-RBF	Agrupados	0.602 (0.029)	**0.606** (0,030)***	21655.71
SVM-RBF	Troceados	0.572 (0.021)	0.576 (0.028)	25297.36
SVM-RBF	11 coordenadas	0.578 (0.019)	0.581 (0.022)	7509.86

Cuadro 4.3: Tasas de acierto promedio (validación cruzada y test) y tiempos de ejecución (en segundos) de cada algoritmo en función de la metodología utilizada para el conjunto de datos sintético con $a = 0{,}1$. Se marcan un *, ** y *** los resultados que son significativamente mejores en el test de Wilcoxon comparados con la metodología original a un nivel de significación del 90 %, 95 % y 99 % respectivamente. En negrita se marca el mejor resultado para cada familia de algoritmos.

Tras realizar este experimento obtenemos las siguientes conclusiones. La primera de ellas es que la familia de algoritmos Random Forest es aquella con la que se obtienen peores tasas de acierto. No obstante estos clasificadores son los más rápidos de entrenar. Además, si ordenamos los clasificadores en función de la tasa de acierto en test, se puede ver cómo tanto RF como las SVM

preservan este orden. Es por esto por lo que RF puede ser una herramienta útil para hacernos una idea a priori de cuál será la mejor metodología a usar en un experimento. Las SVM con núcleo RBF son aquellas con las que se obtienen mejores tasas de acierto en test. Sin embargo también son las más costosas a la hora de elegir la configuración óptima de los hiperparámetros. En cuanto a las metodologías, tanto el trabajar con las componentes principales de las funciones originales (enfoque multivariante) cómo con los coeficientes de Fourier (enfoque funcional) nos aportan resultados mejores que cuando no se aplica ninguna técnica de procesamiento de los datos. Además combinar estos dos enfoques nos permite obtener resultados significativamente mejores (combinar los dos enfoques parece ser una idea razonable). Por otro lado la metodología del troceado de las funciones no parece ser útil. Además, en este experimento, tomar los instantes en los que mejor se separan las esperanzas de las dos clases, nos permite reducir drásticamente el tiempo de ejecución y hace que podamos representar las funciones de una forma sencilla. En definitiva, la mejor manera de trabajar con los datos funcionales, a la vista de los resultados de este experimento, consiste en obtener algunas componentes principales y unos coeficientes de Fourier de las funciones originales y entrenar una SVM con núcleo RBF con este nuevo conjunto de datos.

4.2. Berkeley

El siguiente experimento que vamos a realizar consiste en analizar el conjunto de datos *Berkeley*. En este conjunto de datos se encuentran las mediciones de las alturas de 93 personas (54 mujeres y 39 hombres) en 31 instantes de tiempo, entre los 1 y los 18 años en intervalos no equiespaciados. Nuestro objetivo en este problema es entrenar varios clasificadores utilizando los enfoques multivariante y funcional y ver si aquellos generados con el enfoque funcional obtienen mejores tasas de acierto. Comencemos haciendo un análisis exploratorio de los datos. Para ello, representamos algunas trayectorias y la media más menos una desviación típica según la clase (0 en el caso de los hombres y 1 en el caso de las mujeres). Como puede verse en la Figura 4.27, las alturas de los hombres y las mujeres parece que se comportan de la misma forma en las primeras mediciones, mientras que en las últimas se diferencian mejor. Esto puede deberse a que los hombres tienden a medir más que las mujeres con el tiempo. Hacemos lo mismo con los valores de la primera y segunda derivada. En la Figura 4.2 puede verse cómo las últimas mediciones de las alturas (a partir de la medición número 20) parecen ser útiles a la hora de clasificar un nuevo dato en una de las dos clases. Esto puede deberse a que las mujeres suelen dar el "estirón" antes que los hombres. Este comportamiento parece dar información acerca del género de una persona.

Ahora vamos a modelizar el problema de clasificación utilizando las diferentes metodologías. Para ello vamos a entrenar algunos Random Forest y SVM con diferentes núcleos.

Comenzamos con el conjunto de datos original. El primer algoritmo que vamos a utilizar es Random Forest. Para seleccionar los hiperparámetros `max_features` y `n_estimators` realizamos un estudio de la sensibilidad del acierto. Fijamos el número de atributos `max_features` en 7 (la raíz cuadrada de la cantidad total de variables) y entrenamos 50 RF con diferentes particiones entrenamiento/test para los valores de `n_estimators` en la malla $\{1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047\}$. En la Figura 4.29 podemos ver cómo varía el acierto de test en función de los valores de `n_estimators`. Se puede ver cómo el acierto aumenta según lo hace el número de árboles del RF hasta estancarse. Este patrón se repetirá a lo largo de los experimentos. Tomamos 1023 árboles. Con este valor de τ el acierto se ha saturado y ha dejado de aumentar. Una vez seleccionado el número de árboles del RF vemos si la elección de tomar la raíz cuadrada del número total de atributos (7) es una elección razonable para el hiperparámetro `max_features`. Para ello volvemos a entrenar 50 Random Forest tomando diferentes cantidades del hiperparámetro. Utilizamos la malla $\{1, 2, \ldots, 24\}$. Los resultados pueden verse en la Figura 4.29. Puede verse cómo el acierto crece hasta estancarse a partir de un cierto valor del hiperparámetro. Tomamos $\nu = 6$. Podríamos escoger un valor más pequeño para este hiperparámetro, pero seguimos escogiendo 7 por ser consistentes con la elección estándar. A lo largo de los experimentos, salvo que se especifique lo contrario, tomaremos como valor de `max_features` la raíz cuadrada del número total de atributos.

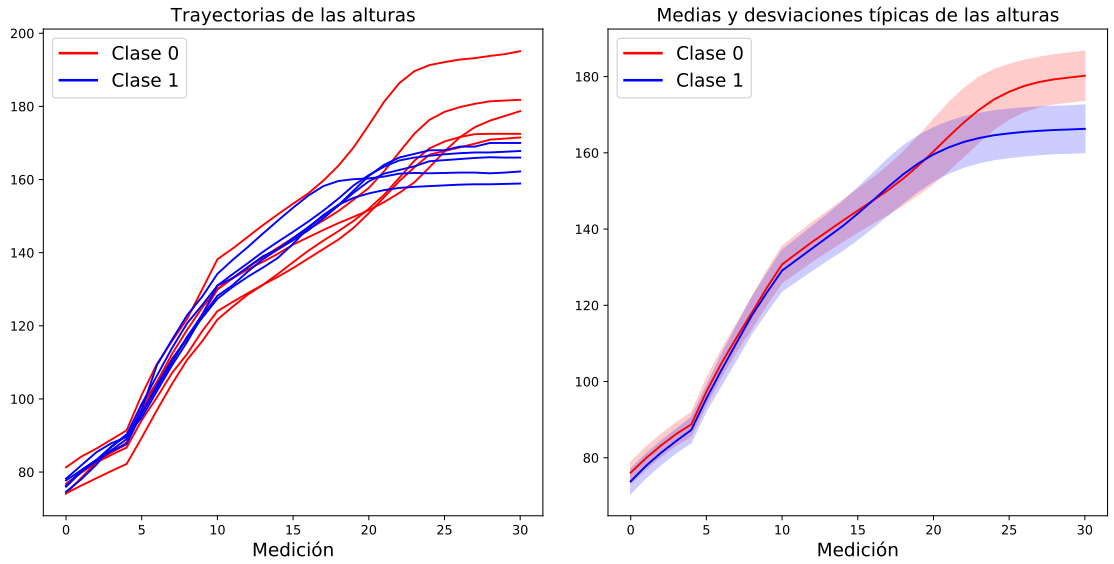

Figura 4.27: A la izquierda pueden verse cinco trayectorias de cada clase. A la derecha, la media de cada clase más menos una desviación típica.

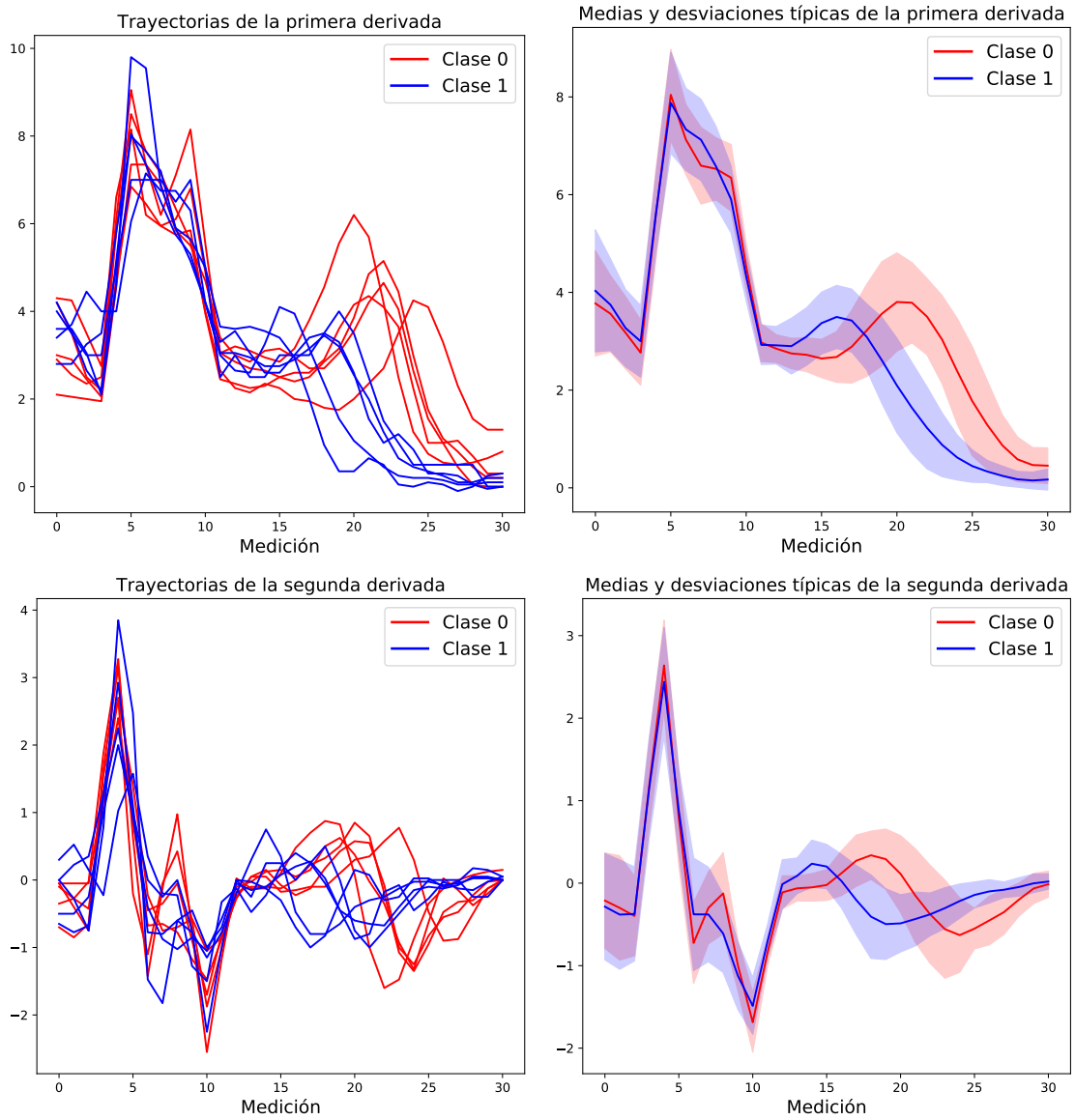

Figura 4.28: En la primera fila de imágenes pueden verse las representaciones correspondientes a la derivada primera, mientras que en la segunda las propias de la derivada segunda. En la primera columna se muestras cinco trayectorias de cada clase, mientras que en la segunda las medias y desviaciones típicas de cada clase.

Con estas configuraciones de los hiperparámetros de Random Forest entrenamos un nuevo clasificador los conjuntos de entrenamiento. Obtenemos una tasa de acierto en test del 96.12 % en 0.68 segundos. Esta tasa de acierto es bastante alta. Esto se debe a que, en rasgos generales, las dos clases son bastante fáciles de separar. Además los tiempos de ejecución son muy bajos.

A continuación entrenamos una SVM con núcleo lineal. Obtenemos el valor óptimo del hiperparámetro `C` mediante una validación cruzada. El valor obtenido ha sido `C`$= 10^{-3}$. Conseguimos un acierto de validación cruzada y test del 96.60 % y 97.02 % respectivamente. Para seleccionar el mejor valor de este hiperparámetro se ha requerido un tiempo de ejecución de 29.44 segundos. En la Figura 4.30 podemos ver cuán sensible es la tasa de acierto de la SVM en función de los valores de `C`. Puede apreciarse cómo a partir de un cierto valor de `C` el acierto crece de manera rápida para mantenerse constante en valores sucesivos.

Aplicando la SVM con núcleo lineal se obtiene una tasa de acierto que mejoran en cerca de un 1 % las de los clasificadores entrenados con Random Forest. Los tiempos de ejecución son muy altos si los comparamos con los requeridos por el Random Forest. Esto se debe a que en el caso de RF no hemos hecho una validación cruzada para seleccionar los valores de los hiperparámetros.

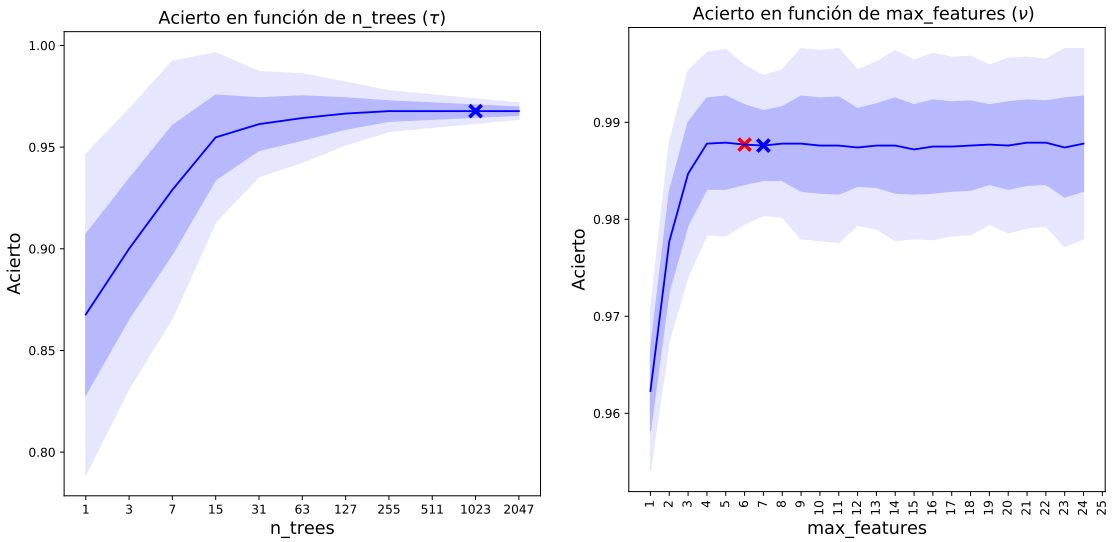

Figura 4.29: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 RF con $\nu = 7$ (izquierda). Se marcan con una cruz los valores de τ escogidos para analizar el conjunto de datos. A la derecha la tasa de acierto promedio en función de $\nu \pm$ una y dos desviaciones típicas en 50 RF con $\tau = 1023$. Se marcan con una cruz en color el valor de ν elegido para el experimento ($\sqrt{31} \sim 6$). En negro se marca $\log_2(31) \sim 5$.

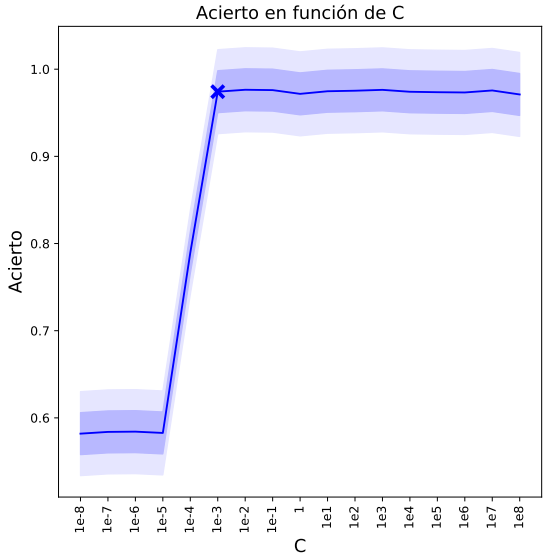

Figura 4.30: Tasa de acierto promedio en función de `C` $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

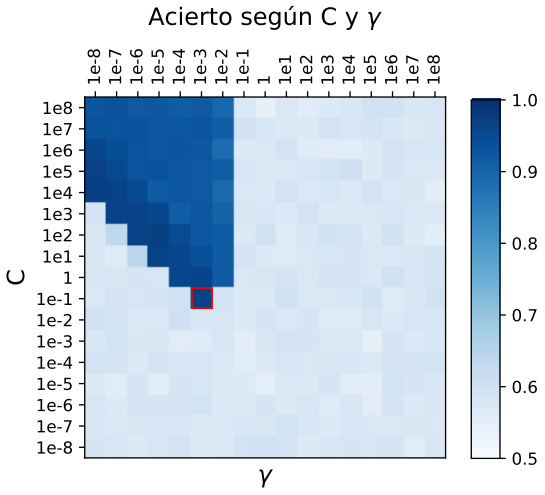

Figura 4.31: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo el valor de los hiperparámetros seleccionado por validación cruzada.

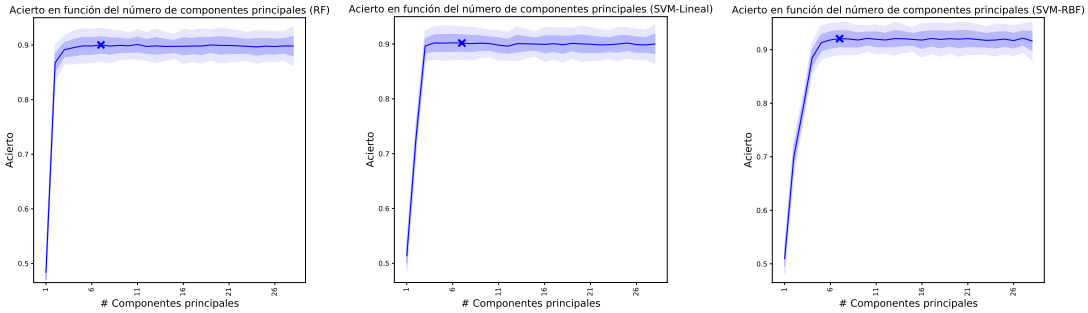

Figura 4.32: Tasa de acierto promedio $\pm$ una y dos desviaciones típicas de 50 Random Forest con $\tau = 1023$ (izquierda), SVM con núcleo lineal (centro) y SVM con núcleo RBF (derecha) en función del número de componentes principales empleadas para generar el conjunto transformado. Se han entrenado las SVM con los valores de los hiperparámetros obtenidos por validación cruzada (para cada número de componentes principales). Se marcan con una cruz la cantidad de componentes principales escogidas (7).

Por último entrenamos una SVM con núcleo RBF. En este caso tenemos que seleccionar la mejor pareja de hiperparámetros C y γ. Los escogemos por validación cruzada. Los valores obtenidos han sido C$= 10^{-1}$ y $\gamma = 10^{-3}$. Para poder ver cómo varían los aciertos en función de los valores de los hiperparámetros mostramos la Figura 4.31. Con esta configuración de los hiperparámetros obtenemos unas tasas de acierto del 96.69 % y 97.28 % en validación cruzada y test respectivamente. Se ha empleado un total de 116.93 segundos en hacer la validación cruzada. Puede verse cómo a partir de ciertos valores del hiperparámetro C la tasa acierto promedio se estanca. Lo contrario ocurre con γ. En este caso el acierto crece hasta que a partir de un cierto valor se desploma. Aplicando el test de Wilcoxon a los resultados con los de la SVM con núcleo lineal obtenemos que no podemos concluir que se aprecien diferencias significativas. Se tiene un $p-$valor de 0.2176, por lo que no podemos rechazar la hipótesis nula. A pesar de obtener este $p-$valor sí que se aprecia un claro aumento en el tiempo de ejecución con respecto a la SVM con núcleo lineal. Esto se debe a que el núcleo RBF requiere seleccionar dos hiperparámetros en lugar de uno.

Pasamos ahora a analizar el segundo método. Para generar el segundo conjunto de datos hemos empleado 7 componentes principales. Tomamos esta cantidad del número de componentes principales porque la tasa de acierto se mantiene constante para valores superiores, cómo puede verse en la Figura 4.32. Podríamos tomar más componentes principales para generar el conjunto de datos transformado pero no aumentaría la tasa de acierto. Una vez tenemos los conjuntos entrenamos los Random Forest. Para hacernos una idea de cómo varían los aciertos en función de los valores

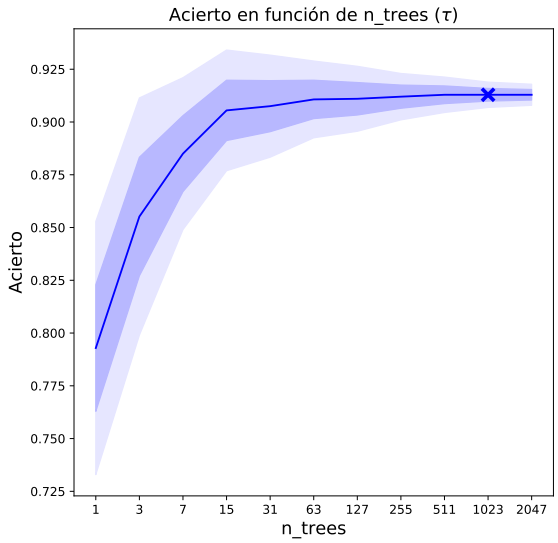

Figura 4.33: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 7$. Se marca con una cruz el valor del hiperparámetro escogido.

del número de árboles del RF (τ) podemos ver la Figura 4.33. Como puede verse tomar como del hiperparámetro `n_estimators` 1023 sigue siendo razonable. Además se puede apreciar cómo la tasa de acierto tiende a un valor asintótico y deja de crecer. Con estas configuraciones de los hiperparámetros obtenemos una tasa de acierto en test del 96.54 %. Se han requerido unos tiempos de ejecución de 0.60 segundos en entrenar el clasificador. Tras entrenar los RF con las componentes principales del conjunto original obtenemos una leve mejoría con respecto a la metodología anterior (de cerca del 0.4 %). Además los tiempos de ejecución se han reducido muy sutilmente. Esto se debe a que la dimensión de los datos se ha reducido. Pasamos ahora a entrenar una SVM con núcleo lineal. Seleccionamos el valor del hiperparámetro `C` por validación cruzada. El valor del hiperparámetro escogido es $\texttt{C} = 10^{-2}$. Las tasas de acierto en validación cruzada y test son del 96.96 % y 97.38 %. Para realizar la validación cruzada se han necesitado 22.80 segundos. En la Figura 4.34 podemos ver cómo varía la tasa de acierto en función del valor del hiperparámetro `C`. El acierto se mantiene muy bajo para valores pequeños de `C` y crece rápidamente a partir de un valor umbral. Después sigue manteniéndose constante.

Con este clasificador se tienen unos resultados mejores que al utilizar la metodología anterior aunque esta diferencia no es lo suficientemente grande cómo para haya una clara evidencia estadística que lo soporte. Aplicando el test de Wilcoxon obtenemos un $p-$valor muy pequeño (0.123). No obstante los tiempos de ejecución empleados para seleccionar las configuraciones de los hiperparámetros se han reducido notablemente. Esto se debe a que se ha disminuido el número de atributos del conjunto. Por último entrenamos una SVM con núcleo RBF. Seleccionamos los valores de los hiperparámetros `C` y γ por validación cruzada. Tenemos que la pareja de hiperparámetros escogida es $\texttt{C} = 10^{-1}$ y $\gamma = 10^{-1}$. Las tasas de acierto en validación cruzada y test son del 97.21 % y 97.61 % en 86.56 segundos. En la Figura 4.35 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros `C` y γ. Para valores grandes de γ el error es muy alto. Lo mismo ocurre para valores pequeños de `C`. Sin embargo, cuando `C` es grande y γ pequeño la tasa de acierto es alta. Las mejores tasas de acierto se obtienen en un valor intermedio. Este valor coincide con las configuraciones óptimas de los hiperparámetros halladas por validación cruzada.

A pesar de obtener mejores tasas de acierto que al entrenar los mismos clasificadores con el conjunto original, aplicando el test de Wilcoxon no obtenemos evidencia estadística suficiente como para afirmar que existe una diferencia significativa. No obstante sí se puede apreciar cómo los tiempos de ejecución se han reducido en gran medida. Esto se debe a que hemos reducido notablemente el número de atributos. A continuación utilizamos el tercer conjunto de datos. Para ello hemos proyectado las funciones originales en la base de Fourier con 3 elementos. Seleccionamos esta cantidad del parámetro n_F por validación cruzada usando Random Forest (en las Figuras 5.5 y 5.6 de la Sección 5.2.2 del apéndice pueden verse las gráficas de las elecciones por validación cruzada

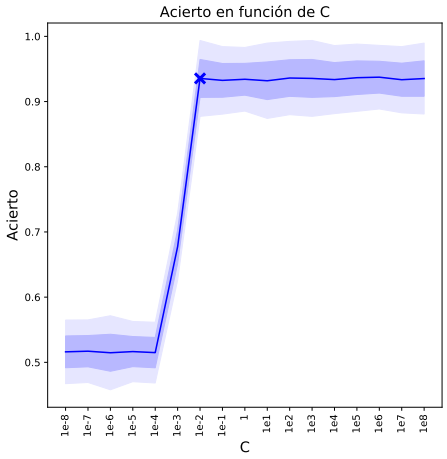

Figura 4.34: Tasa de acierto promedio en función de C ± una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

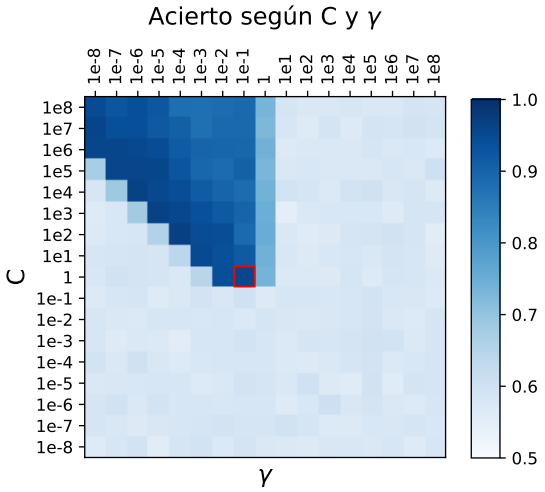

Figura 4.35: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo valores de los hiperparámetros escogidos por validación cruzada.

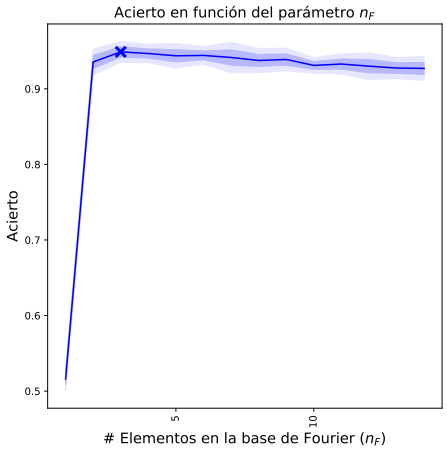

Figura 4.36: Tasa de acierto promedio ± una y dos desviaciones típicas de 50 Random Forest en función del número de elementos de la base de Fourier (n_F) empleados para generar el conjunto transformado. Se marca con una cruz el valor de n_F escogido por validación cruzada.

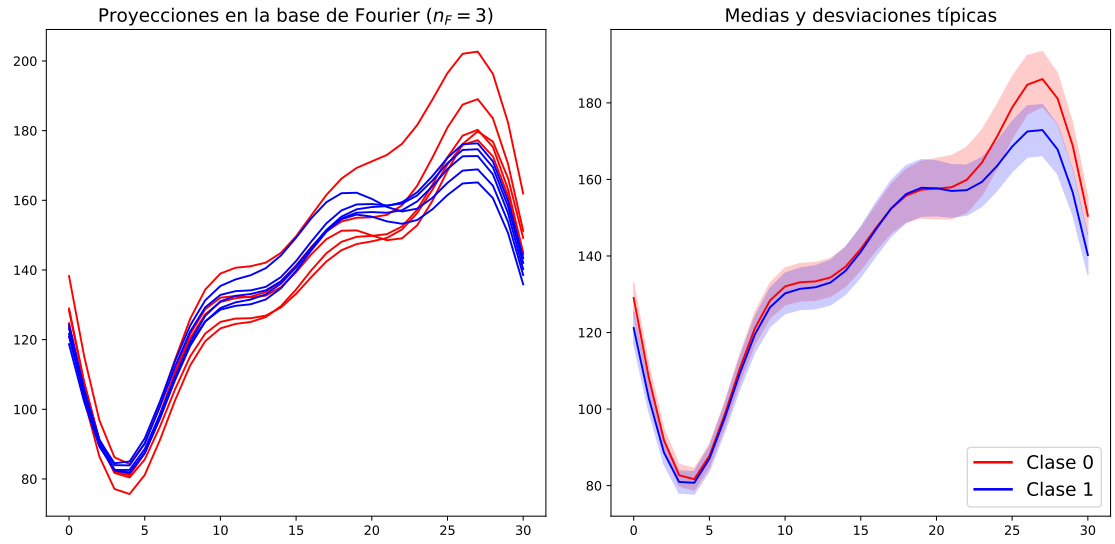

Figura 4.37: A la izquierda, las proyecciones de 5 funciones de cada clase del conjunto original en la base de Fourier con 3 elementos ($n_F = 3$). A la derecha las medias de cada clase $\pm$ una desviación típica de todo el conjunto proyectado en esta base.

para los clasificadores SVM con núcleos lineal y RBF). Cómo puede apreciarse en la Figura 4.36, proyectar las funciones en bases de Fourier con más elementos hace que disminuya la tasa de acierto (se añade ruido). Este acierto crece según aumenta el número de elementos en la base de Fourier hasta un cierto valor umbral. Después decrece lentamente según aumenta el parámetro n_F.

Para hacernos una idea de que aspecto tienen las funciones proyectadas en la base de Fourier, mostramos cinco funciones de cada clase. En la Figura 4.37 podemos ver cómo las funciones se han suavizado y parecen separarse ligeramente mejor en los primeros instantes de tiempo. Esto nos hace pensar que, con este conjunto de datos, se obtendrán mejores tasas de acierto que con el conjunto original. Entrenamos los Random Forest. Para ello tomamos 1023 como valor del hiperparámetro `n_estimators`. En la Figura 4.38 podemos ver cómo esta elección sigue siendo razonable. Además aquí podemos observar cómo varían los aciertos en función de los valores de τ. La tasa de acierto crece según lo hace τ hasta estancarse en un valor asintótico. Obtenemos una tasa de acierto del 95.70 % en test en un total de 0.60 segundos. Podemos observar una leve mejoría con respecto a los aciertos conseguidos usando el conjunto de datos original y peores que cuando aplicamos la metodología de las componentes principales. Además, a pesar de trabajar con 15 atributos podemos representar las funciones proyectadas para hacernos una idea de su aspecto (véase la Figura 4.38). Esto es una clara ventaja con respecto a la metodología de las componentes principales (en este caso sólo podemos visualizar las transformaciones cuando se utilizan 2 o 3 componentes principales).

Ahora entrenamos una SVM con núcleo lineal. Escogemos el valor del hiperparámetro `C` por validación cruzada. El valor del hiperparámetro seleccionado es `C` $= 10^{-2}$. Las tasas de acierto en validación cruzada y test son del 96.02 % y 96.49 % respectivamente. Se han empleado 19.92 segundos en realizar la selección de los hiperparámetros por validación cruzada. En la Figura 4.39 podemos ver cómo varía la tasa de acierto en función de los diferentes valores del hiperparámetro `C` de la malla. La tasa de acierto es baja para valores pequeños de `C`. Llegado un cierto valor del hiperparámetro crece muy rápidamente para mantenerse constantes para valores superiores.

Con las configuraciones de los hiperparámetros escogidos por validación cruzada obtenemos resultados similares a los de los clasificadores entrenados usando la metodología de las componentes principales, aunque ligeramente inferiores. No obstante, no se consigue evidencia estadística suficiente en el test de Wilcoxon que apoye esta afirmación con un nivel de significación suficientemente grande.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros `C` y γ usamos una validación cruzada. La pareja de hiperparámetros escogida es `C` $= 10^{-1}$ y $\gamma = 10^{-1}$. En la Figura 4.40 podemos ver cómo varían las tasas de acierto en función cada pareja de hiperparámetros `C` y γ. Los aciertos se mantienen altos para valores grandes de `C` y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo.

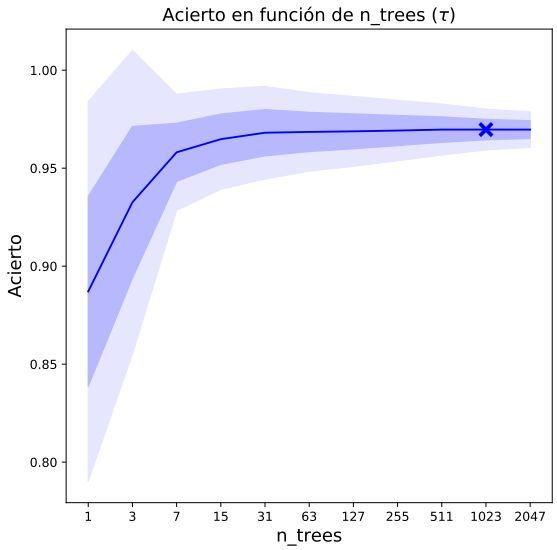

Figura 4.38: Tasa de acierto promedio en función de τ $\pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 7$. Se marca con una cruz el valor del hiperparámetro seleccionado.

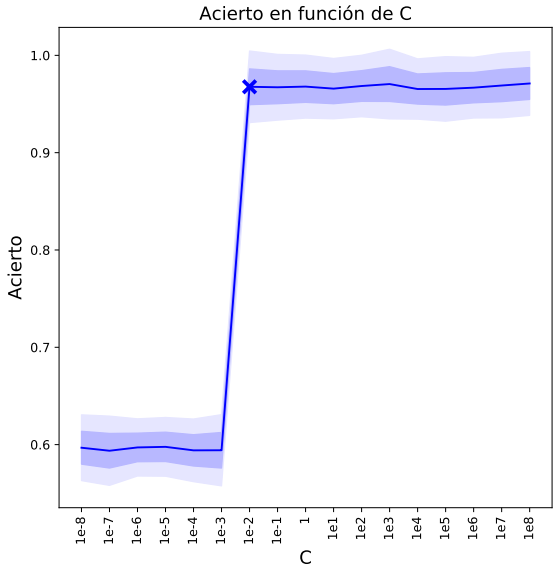

Figura 4.39: Tasa de acierto promedio en función de `C` $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogidos por validación cruzada.

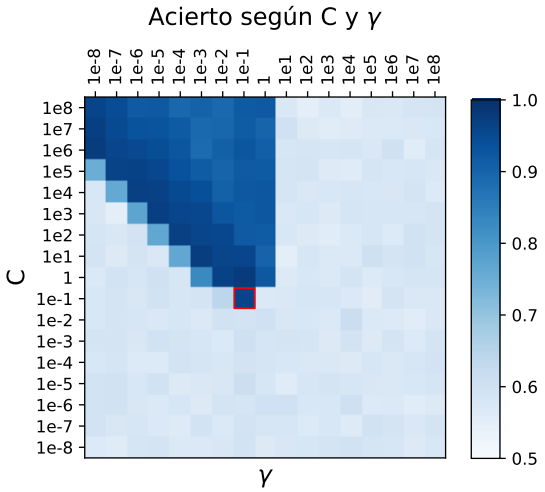

Figura 4.40: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

Estos valores coinciden con los escogidos por validación cruzada. Las tasas de acierto de validación cruzada y test son del 96.47 % y 96.80 %. Se han empleado 80.70 segundos en realizar la selección de la pareja de hiperparámetros.

Los resultados obtenidos son muy similares a los que se tienen cuando utilizamos las SVM con núcleo lineal. Las tasas de acierto son muy parecidas a las que se tienen cuando entrenamos los clasificadores aplicando la metodología de las componentes principales. Tampoco se obtiene evidencia estadística suficiente para poder afirmar, aplicando el test de Wilcoxon, que hay un claro empeoramiento con respecto a la metodología original.

A continuación juntamos los dos conjuntos de datos anteriores (las componentes principales y los coeficientes en las bases de Fourier). Con el conjunto agrupado entrenamos un Random Forest. Utilizamos el mismo valor del hiperparámetro τ. En la Figura 4.41 podemos observar cómo varían los aciertos en función de τ. También podemos ver cómo tomar $\tau = 1023$ sigue siendo una elección razonable. Puede apreciarse el mismo fenómeno que con los Random Forest anteriores. El acierto crece con τ hasta un valor asintótico. Obtenemos una tasa de acierto en test del 98.7 %. Se ha necesitado un total de 0.62 segundos para entrenar el clasificador.

Es con este conjunto de datos con el que se consigue la mejor tasas de acierto de entre todos los clasificadores Random Forest. La tasa de acierto aumentan en cerca de un 2.5 % con respecto a los resultados de la metodología original. A pesar de no parecer especialmente útiles los coeficientes de Fourier de las funciones originales, al juntarlos con las componentes principales, el acierto crece.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro C por validación cruzada. Tenemos que el valor del hiperparámetro escogido es $C = 10^{-2}$. En la Figura 4.42 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro C. Se tiene un error alto para valores pequeños de C. Este decrece rápidamente hasta mantenerse constante en valores de C superiores. Las tasa de acierto en validación cruzada y test es de 97.8 % y 98.15 % en 23.19 segundos respectivamente.

Es aplicando esta metodología cuando obtenemos las mejores tasas de acierto de entre todas las SVM con núcleo lineal. Además obtenemos resultados significativamente mejores a un nivel de significación del 95 % si lo comparamos con los resultados de entrenar el clasificador con los conjuntos originales. Aplicando el test de Wilcoxon obtenemos que el $p-$valor del contraste vale 0,026. Además los tiempos requeridos para seleccionar las configuraciones de los hiperparámetros es bastante menor.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros C y γ usamos una validación cruzada. Tenemos que la pareja de hiperparámetros escogida es $C = 10^{-1}$ y $\gamma = 10^{-1}$. En la Figura 4.43 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Se puede apreciar el mismo comportamiento que en las SVM con núcleo RBF anteriores. Los aciertos son altos para valores grandes de C y pequeños de γ. en cualquier otro caso los aciertos son muy bajos. Hay un valor intermedio de los hiperparámetros

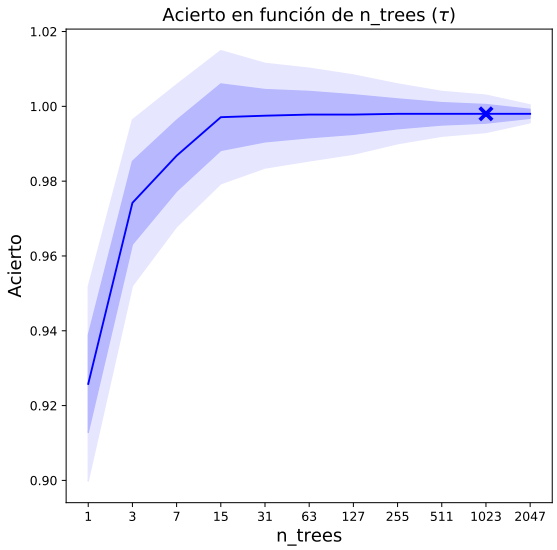

Figura 4.41: Tasa de acierto promedio en función de τ $\pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 7$. Se marca con una cruz el valor del hiperparámetro seleccionado.

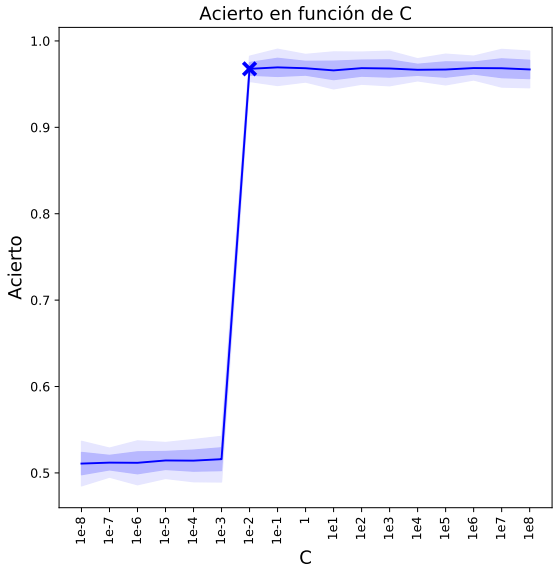

Figura 4.42: Tasa de acierto promedio en función de `C` $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

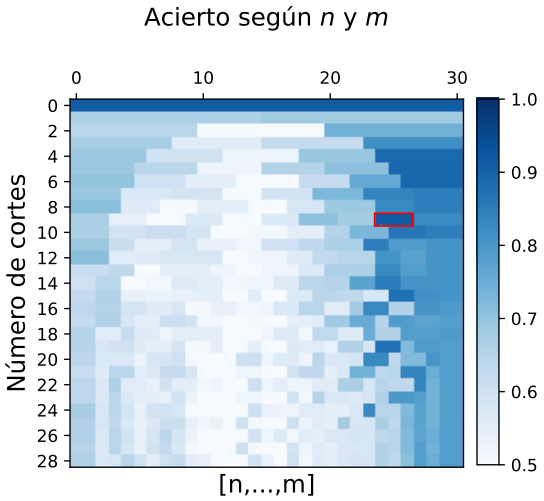

Figura 4.44: Tasas de acierto promedio de 50 Random Forest para los conjuntos troceados por los instantes $[n, \ldots, m]$.

donde el error es mínimo. Este valor intermedio coincide con las configuraciones obtenidas mediante validación cruzada. Las tasas de acierto en validación cruzada y test son del 98.8 % y 99.03 % en 94.05 segundos.

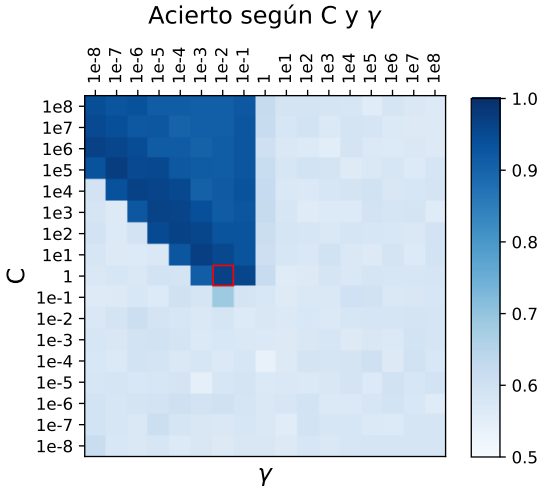

Figura 4.43: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

Usando esta metodología obtenemos las mejores tasas de acierto de todo el experimento. Los resultados son significativamente mejores que cuando se aplica el conjunto de datos original para entrenar los clasificadores para cualquier nivel de significación habitual. Aplicando el test de Wilcoxon obtenemos que el $p-$valor del contraste vale 0,0006. Por lo tanto hay evidencia estadística suficiente para afirmar que las tasas de acierto de los clasificadores entrenados usando la metodología de juntar las componentes principales y los coeficientes de Fourier son mejores que los obtenidos utilizando el conjunto original.

Pasamos a utilizar la técnica del troceado de las funciones siguiendo la metodología explicada en la Sección 3.3. Troceamos las funciones originales en varias subfunciones y entrenamos 50 Random Forest para seleccionar los mejores lugares por donde cortar las funciones originales. En la Figura 4.44 podemos ver las tasas de acierto promedio para cada lugar de corte de las funciones originales.

Puede verse cómo, según se toman intervalos más pequeños el acierto disminuye. Además el acierto es mayor para valores grandes de n y m (el final de las funciones originales parece dar

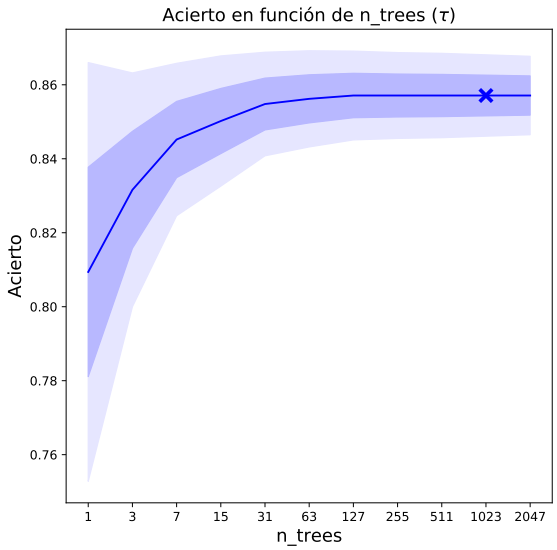

Figura 4.45: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 7$. Se marca con una cruz el valor del hiperparámetro seleccionado.

más información que el principio, como es de esperar). La mejor tasa de acierto se tiene cuando se trocean las funciones originales por los instantes $[24, 25, 26]$. Por lo tanto los valores de n y m que utilizamos para generar las subfunciones del conjunto de datos troceado son $n = 24$ y $m = 26$.

Una vez hemos troceado las funciones entrenamos un Random Forest. Utilizamos los mismos valores del hiperparámetro τ que en los casos anteriores. En la Figura 4.45 podemos observar cómo varía la tasa de acierto en función de los valores de τ de la malla. Además se puede apreciar cómo la elección de $\tau = 1023$ sigue siendo razonable. La tasa de acierto crece según lo hace τ hasta estancarse en un valor asintótico. Se obtiene una tasa de acierto del 93.3 % en test. Se ha empleado un total de 0.61 segundos.

Con esta metodología de trocear las funciones originales obtenemos los peores resultados de todo el experimento. Estos resultados son significativamente peores que los obtenidos con la metodología original para cualquier nivel de significación. Obtenemos un $p-$valor usando el test de Wilcoxon de 10^{-3}. No obstante el tiempo de ejecución empleado en entrenar los clasificadores se reduce levemente. Esto se debe a que se disminuye a la mitad el número de atributos de los datos.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro `C` por validación cruzada. Para cada valor de a tenemos que el valor del hiperparámetro escogido es $\texttt{C} = 10^{-3}$. En la Figura 4.46 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro `C`. La tasa de acierto es baja para valores pequeños de `C`. Llegado un cierto valor del hiperparámetro crece muy rápidamente para mantenerse constantes para valores superiores. La tasa de acierto de validación cruzada y test es del 94.0 % y 94.4 % en 20.25 segundos.

Los aciertos de las SVM entrenados a partir de los datos originales troceados también obtienen las peores tasas de acierto de entre todas las metodologías utilizadas. Si los comparamos con la tasa de acierto del clasificador entrenado a partir del conjunto original, obtenemos evidencia estadística suficiente (en el test de Wilcoxon) para afirmar que, a un nivel de significación del 95 %, se obtienen peores resultados. El $p-$valor en este caso vale 0,015.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros `C` y γ usamos una validación cruzada. Para cada valor de a tenemos que la pareja de hiperparámetros escogida es $\texttt{C} = 10^{-1}$ y $\gamma = 10^{-2}$. En la Figura 4.47 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros `C` y γ. Los aciertos se mantienen altos para valores grandes de `C` y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. La tasa de acierto promedio de validación cruzada y test es del 94.3 % y 94.7 % en 80.49 segundos.

Las tasas de acierto de estos clasificadores son peores que los que se obtienen al utilizar el conjunto de datos original. Aplicando el test de Wilcoxon podemos afirmar a un nivel de significación

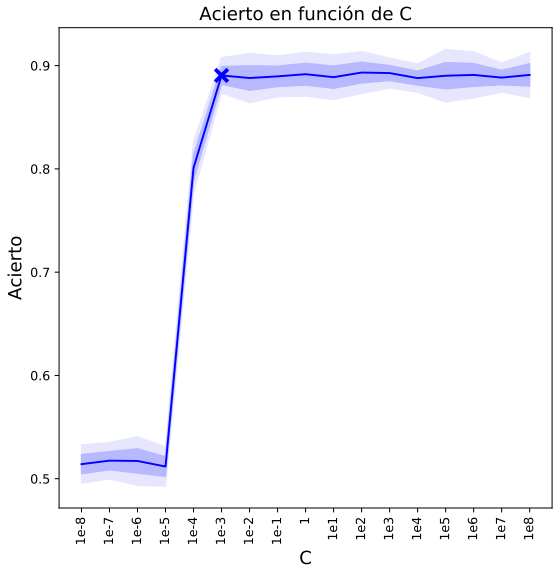

Figura 4.46: Tasa de acierto promedio en función de C ± una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

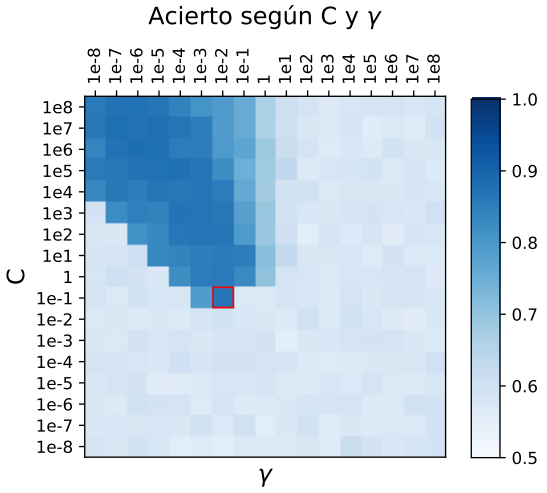

Figura 4.47: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo el valor de los hiperparámetros escogido por validación cruzada.

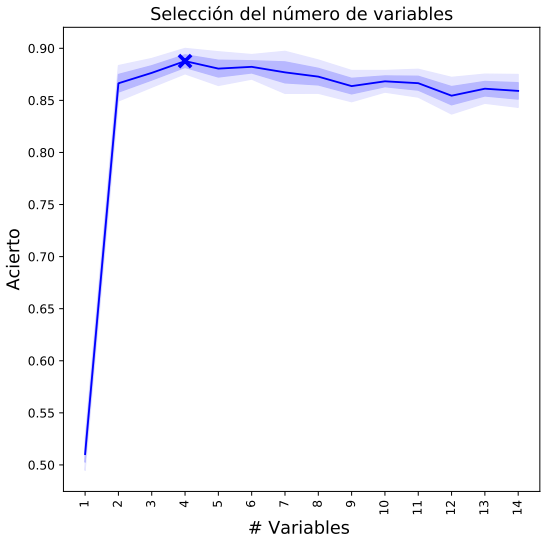

Figura 4.48: Tasas de acierto promedio en test y validación cruzada ± una y dos desviaciones típicas en función del número de variables seleccionadas. Se marcan con una cruz el número de variables seleccionadas por validación cruzada.

del 95 % que los resultados son peores que los que se obtienen aplicando la metodología original. En este caso el $p-$valor del contraste vale 0,017.

Por último reducimos la dimensión de los datos mediante una selección de variables de los instantes de las funciones. Aplicamos la metodología explicada en la Sección 3.3 y usando Random Forest. De esta manera obtenemos que vamos a seleccionar 4 variables (instantes temporales). En las Figuras 5.7 y 5.8 de la Sección 5.2.2 del apéndice pueden verse las gráficas de las elecciones por validación cruzada para los clasificadores SVM con núcleos lineal y RBF. En la Figura 4.48 podemos ver las tasas de acierto de validación cruzada.

Entrenamos un Random Forest. Utilizamos los mismos valores del hiperparámetro τ que en los casos anteriores. En la Figura 4.49 podemos observar cómo varía la tasa de acierto en función de los valores de τ de la malla. Además se puede apreciar cómo las elecciones de $\tau = 1023$ siguen siendo adecuadas. La tasa de acierto crece según lo hace τ hasta estancarse en un valor asintótico. De esta forma se obtiene una tasa de acierto en test del 95.1 %. Se ha requerido un total de 0.65 segundos.

Utilizando esta metodología para entrenar los Random Forest obtenemos una tasa de acierto ligeramente inferior que cuando usamos todo el conjunto original. Aunque no obtenemos evidencia estadística suficiente para concluir que los resultados son mejores a un nivel de significación elevado, sí podemos observar que los tiempos de ejecución han disminuido.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro `C` por validación cruzada. Tenemos que el valor del hiperparámetro escogido es $\texttt{C} = 10^{-3}$. En la Figura 4.50 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro `C`. Las tasas de acierto son bajas para valores pequeños de `C`. Llegado un cierto valor del hiperparámetro crecen muy rápidamente para mantenerse constantes para valores superiores. Las tasas de acierto de validación cruzada y test es del 95.9 % y 96.3 % en 20.86 segundos .

Los resultados que se obtienen al utilizar las SVM con núcleo lineal son similares a los que se consiguen con los Random Forest. Las tasas de acierto son ligeramente inferiores y los tiempos de ejecución empleados en hallar las mejores configuraciones de los hiperparámetros son mucho menores.

Por último entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros `C` y γ usamos una validación cruzada. Tenemos que la pareja de hiperparámetros escogida es $\texttt{C} = 10^{-1}$ y $\gamma = 10^{-2}$. En la Figura 4.51 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros `C` y γ. Los aciertos se mantienen altos para valores grandes de `C` y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. Las tasas de acierto de validación cruzada y test son del 96.0 % y 96.4 % en 78.69 segundos.

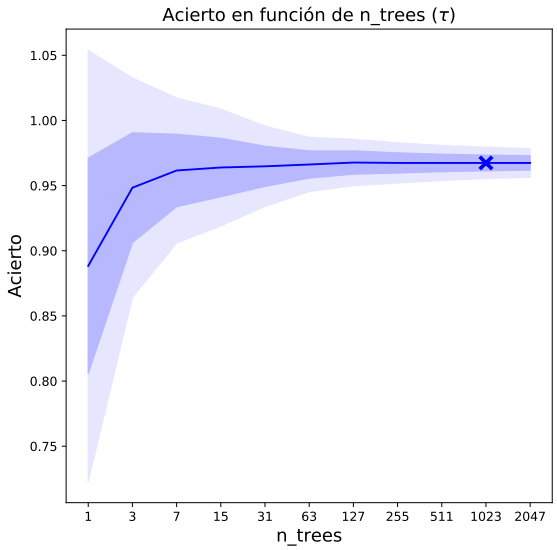

Figura 4.49: Tasa de acierto promedio en función de τ $\pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 7$. Se marca con una cruz el valor del hiperparámetro seleccionado.

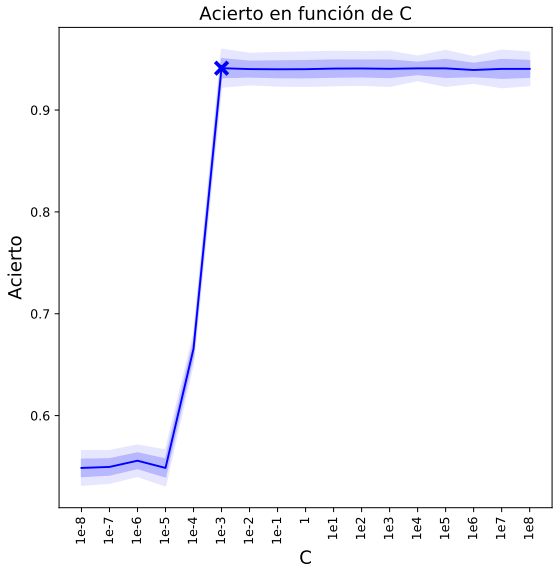

Figura 4.50: Tasa de acierto promedio en función de C $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

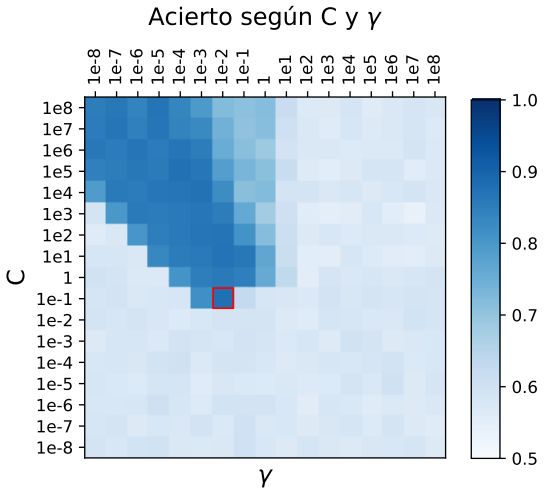

Figura 4.51: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

En este caso obtenemos resultados ligeramente inferiores que cuando utilizamos la metodología original para entrenar los clasificadores. No se obtiene evidencia estadística suficiente para concluir esta afirmación a un nivel de significación del 99 % pero sí se puede apreciar una clara mejoría con respecto a las SVM con núcleo lineal.

A modo de resumen, mostramos a continuación una tabla con los resultados del experimento.

Algoritmo	Método	$\mu_{CV}(\sigma_{CV})$	$\mu_{test}(\sigma_{test})$	t
RF	Original	X	0.961	0.68
RF	7 componentes principales	X	0.965	0.60
RF	Fourier ($n_F = 3$)	X	0.957	0.60
RF	Agrupados	X	**0.987**	0.62
RF	Troceados	X	0.933	0.61
RF	4 coordenadas	X	0.968	0.66
SVM-Lineal	Original	0.966 (0.046)	0.970 (0.041)	29.44
SVM-Lineal	7 componentes principales	0.969 (0.035)	0.973 (0.030)	22.80
SVM-Lineal	Fourier ($n_F = 4$)	0.960 (0.047)	0.964 (0.038)	19.92
SVM-Lineal	Agrupados	0.978 (0.035)	**0.981** (0,032)**	23.19
SVM-Lineal	Troceados	0.940 (0.042)	0.944 (0.038)	20.25
SVM-Lineal	4 coordenadas	0.970 (0.042)	0.972 (0.035)	20.98
SVM-RBF	Original	0.966 (0.040)	0.972 (0.035)	116.93
SVM-RBF	7 componentes principales	0.972 (0.039)	0.976 (0.050)	86.56
SVM-RBF	Fourier ($n_F = 3$)	0.964 (0.039)	0.968 (0.046)	80.70
SVM-RBF	Agrupados	0.988 (0.040)	**0.990** (0,035)***	94.05
SVM-RBF	Troceados	0.943 (0.004)	0.947 (0.030)	80.49
SVM-RBF	4 coordenadas	0.968 (0.046)	0.971 (0.039)	78.26

Cuadro 4.4: Tasas de acierto promedio y tiempos de ejecución de cada algoritmo en función de la metodología utilizada para el experimento `berkeley`. Se marcan un *, ** y *** los resultados que son significativamente mejores en el test de Wilcoxon comparados con la metodología original a un nivel de significación del 90 %, 95 % y 99 % respectivamente. En negrita se marca el mejor resultado para cada familia de algoritmos.

A la vista de los resultados del experimento podemos sacar varias conclusiones. La primera de ellas es que, en general, todos los clasificadores se comportan de manera razonable (prácticamente todos obtienen tasas de acierto promedio superiores al 90 %). Esto nos lleva a pensar que los datos del conjunto Berkeley son fáciles de clasificar. También hemos de tener en cuenta que este conjunto de datos es relativamente pequeño. Contamos con un total de 93 datos funcionales, de los cuales

solo utilizamos 31 para evaluar los clasificadores (que han sido entrenados con los 62 restantes). Esto hace que una mala clasificación de un dato del conjunto de test resulte en una disminución de más del 3 % en la tasa de acierto (esta disminución es incluso más grande en la validación). Si ordenamos los clasificadores según la mejor tasa de acierto en test obtenemos el mismo orden que con el experimento de los datos sintéticos. RF es el clasificador que obtiene peores resultados seguido de las SVM con núcleo lineal. Las SVM con núcleo RBF son aquellas que consiguen mejores tasas de acierto en test. Si nos centramos en los tiempos de ejecución el orden se invierte.

Si nos centramos en los resultados de los Random Forest podemos ver que el tiempo de ejecución empleado para entrenar los clasificadores es muy bajo en todas las metodologías. Esto se debe a que los hiperparámetros `n_estimators` y `max_features` se escogieron previamente y no fue necesario realizar una validación cruzada. En cuanto a los aciertos destaca frente al resto el de la metodología del agrupamiento de las componentes principales y los coeficientes de Fourier. Por otro lado, la técnica con la que se consigue el peor acierto es la del troceado de las funciones. Esto mismo ocurría en el experimento de los datos sintéticos.

En cuanto a las tasas de acierto obtenidas por las SVM con núcleo lineal podemos observar cómo la mejor metodología vuelve a ser la del agrupamiento de las componentes principales y de los coeficientes de Fourier. Aplicando el test de Wilcoxon se obtiene evidencia estadística al 95 % para afirmar que esta metodología es mejor que la original (el $p-$valor del contraste vale $26 \cdot 10^{-3}$). Por el contrario, la metodología que obtiene las peores tasas de acierto vuelve a ser la del troceado de las funciones. Si comparamos la tasa de acierto con la del clasificador entrenado usando el conjunto original se obtiene evidencia estadística suficiente para afirmar que, a un nivel de significación del 95 %, esta aporta peores resultados (el $p-$valor en este caso vale $15 \cdot 10^{-3}$). A pesar de esto, el tiempo empleado en seleccionar los hiperparámetros de la SVM es de los más bajos (esto se puede deber a que se ha reducido drásticamente el número de atributos al trocear las funciones). Cabe destacar que aunque los resultados de la metodología de la proyección de las funciones en la base de Fourier no sea la que consigue la mejor tasa de acierto, si es aquella en la que la selección del hiperparámetro de la SVM emplea menos tiempo.

Con respecto a las SVM con núcleo RBF se obtienen conclusiones parecidas a las anteriores. La metodología con la que se consigue una mejor tasa de acierto en test vuelve a ser la del agrupamiento de las componentes principales y los coeficientes de Fourier. En este caso, aplicando el test de Wilcoxon, se obtiene evidencia estadística suficiente para afirmar que los resultados al aplicar esta metodología son significativamente mejores, si los comparamos con los de la metodología original, a cualquier nivel de significación habitual. En este caso el $p-$valor del contraste vale $6 \cdot 10^{-4}$. Además el tiempo de ejecución empleado para seleccionar los valores de los hiperparámetros es menor. Por el contrario, la metodología que obtiene peores tasas de acierto vuelve a ser la del troceado de las funciones. Comparándolo con la metodología original y aplicando el test de Wilcoxon se obtiene obtiene un $p-$valor de $17 \cdot 10^{-3}$, por lo que hemos conseguido evidencia estadística suficiente para afirmar que los resultados son peores a un nivel de significación del 95 %.

Podemos concluir que la mejor metodología para clasificar los datos funcionales en este experimento consiste en agrupar las componentes principales y los coeficientes de Fourier de los datos, mientras que la peor (en términos de acierto en test) es la del troceado de las funciones. El resto de metodologías, a pesar de no ser las que consiguen mejores tasas de acierto sí parecen ser técnicas útiles para reducir el número de atributos de los datos.

4.3. Phoneme

El siguiente conjunto de datos que vamos a analizar es el conocido como "phoneme". Este conjunto contiene los log-peridiogramas de varios marcos de voz. Cada marco de voz está representado por 152 observaciones a un ratio de muestreo de 16 kHz. Cada dato puede pertenecer a una de las cinco posibles clases ("aa", "ao", "dcl", "iy" o "sh"), pero nosotros vamos a considerar solo las clases "aa" y "ao" (0 y 1 respectivamente), que son aquellas que se separan con mayor dificultad. Cada una de estas clases contiene un total de 400 log-peridiogramas. El objetivo de este experimento es aplicar las metodologías expuestas en la Sección 3.3 y analizar cuál de estas consigue una mejor tasa de acierto en test. Comenzamos haciendo un análisis descriptivo de los datos. Algunas trayectorias de cada clase y las medias y desviaciones típicas pueden verse en la Figura 4.52

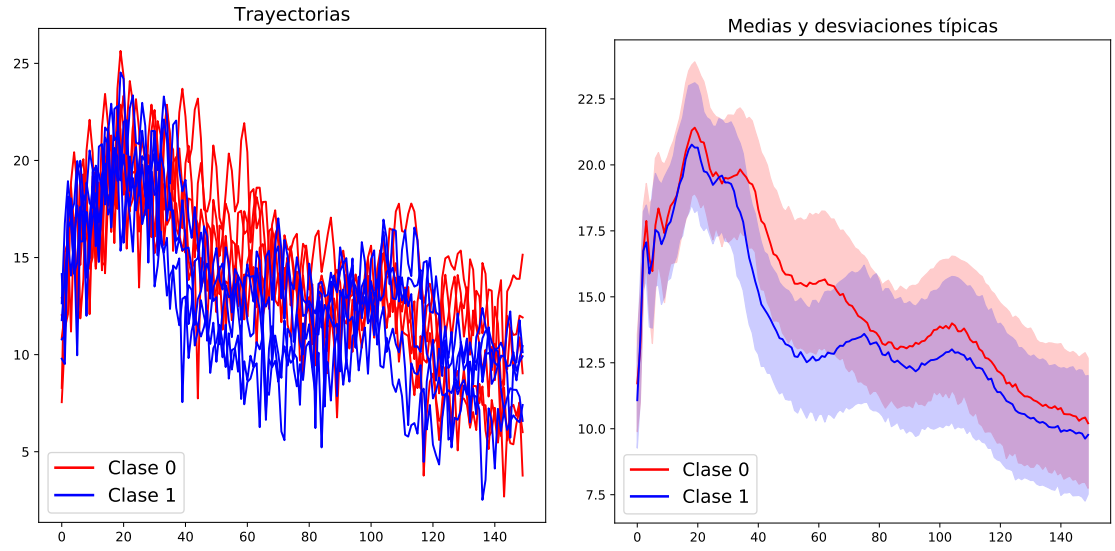

Figura 4.52: A la izquierda 5 trayectorias de cada una de las clases. A la derecha, sus medias y desviaciones típicas.

Se puede ver en la Figura 4.52, parece que las mediciones entre la 40 y la 70 pueden aportarnos información útil a la hora de clasificar una nueva observación, aunque ambas clases están muy juntas, por lo que no esperamos obtener tasas de acierto muy altas. Ahora vamos a modelizar el problema de clasificación utilizando las diferentes metodologías. Para ello vamos a entrenar algunos Random Forest y SVM con diferentes núcleos. Comenzamos con el conjunto de datos original. El primer algoritmo que vamos a utilizar es Random Forest. Para seleccionar los hiperparámetros `max_features` y `n_estimators` realizamos un estudio de la sensibilidad del acierto. Fijamos el número de atributos `max_features` en 12 (la raíz cuadrada de la cantidad total de variables) y entrenamos 50 RF con diferentes particiones entrenamiento/test para los valores de `n_estimators` en la malla $\{1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047\}$. En la Figura 4.53 podemos ver cómo varía el acierto de test en función de los valores de `n_estimators`. Se puede ver cómo el acierto aumenta según lo hace el número de árboles del RF hasta estancarse. Este patrón se repetirá a lo largo de los experimentos. Tomamos 1023 árboles. Con este valor de τ el acierto se ha saturado y ha dejado de aumentar. Una vez seleccionado el número de árboles del RF vemos si la elección de tomar la raíz cuadrada del número total de atributos (12) es una elección razonable para el hiperparámetro `max_features`. Para ello volvemos a entrenar 50 Random Forest tomando diferentes cantidades del hiperparámetro. Utilizamos la malla $\{1, 2, \ldots, 24\}$. Los resultados pueden verse en la Figura 4.53. Puede verse cómo el acierto crece hasta estancarse a partir de un cierto valor del hiperparámetro. Tomamos $\nu = 12$. Podríamos escoger un valor más pequeño para este hiperparámetro, pero seguimos escogiendo 12 por ser consistentes con la elección estándar. A lo largo de los experimentos, salvo que se especifique lo contrario, tomaremos como valor de `max_features` la raíz cuadrada del número total de atributos.

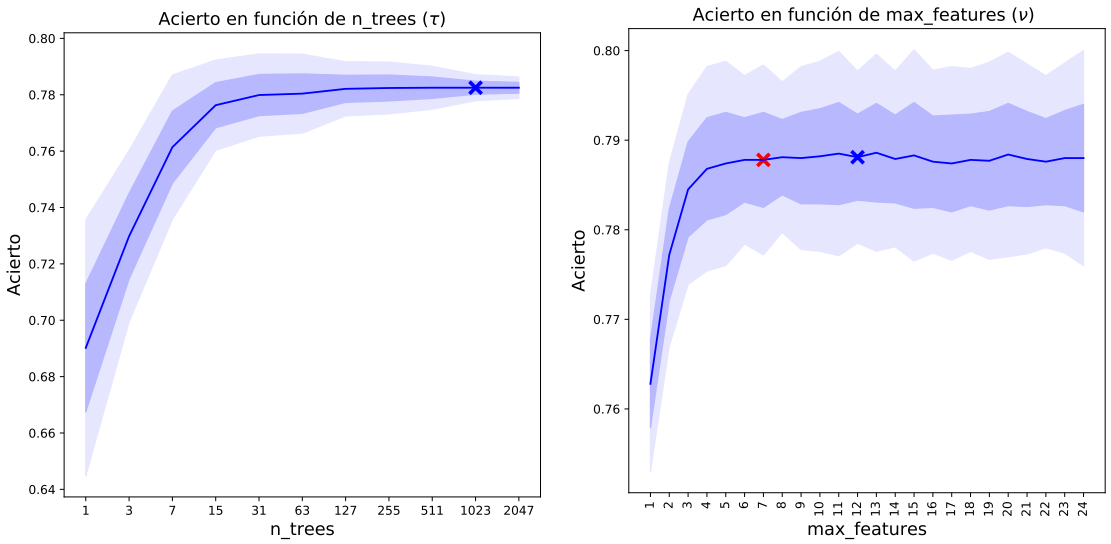

Figura 4.53: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 RF con $\nu = 12$ (izquierda). Se marcan con una cruz los valores de τ escogidos para analizar el conjunto de datos. A la derecha la tasa de acierto promedio en función de $\nu \pm$ una y dos desviaciones típicas en 50 RF con $\tau = 1023$. Se marcan con una cruz en color el valor de ν elegido para el experimento ($\sqrt{152} \sim 12$). En negro se marca $\log_2(152) \sim 7$.

Con esta configuración de los hiperparámetros de Random Forest entrenamos un nuevo clasificador los conjuntos de entrenamiento. Obtenemos una tasa de acierto en test del 78.3 % en 1.21 segundos. Esta tasa de acierto es razonable. Además los tiempos de ejecución son muy bajos.

A continuación entrenamos una SVM con núcleo lineal. Obtenemos el valor óptimo del hiperparámetro `C` mediante una validación cruzada. El valor obtenido ha sido `C`$= 10^{-4}$. Conseguimos un acierto de validación cruzada y test del 80.1 % y 80.6 % respectivamente. Para seleccionar el mejor valor de este hiperparámetro se ha requerido un tiempo de ejecución de 29.44 segundos. En la Figura 4.54 podemos ver cuán sensible es la tasa de acierto de la SVM en función de los valores de `C`. Puede apreciarse cómo a partir de un cierto valor de `C` el acierto crece de manera rápida para mantenerse constante en valores sucesivos.

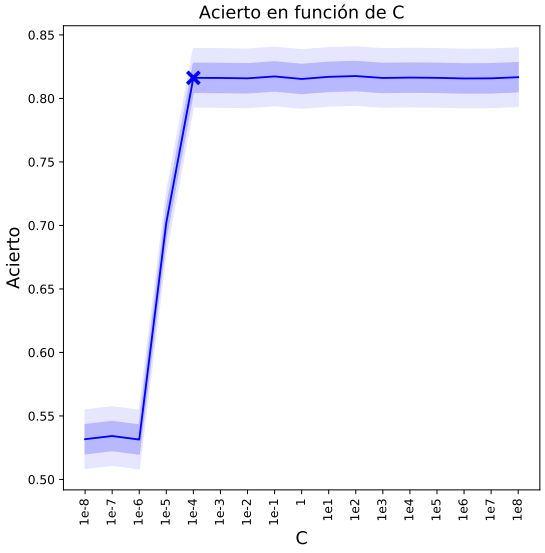

Figura 4.54: Tasa de acierto promedio en función de `C` $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

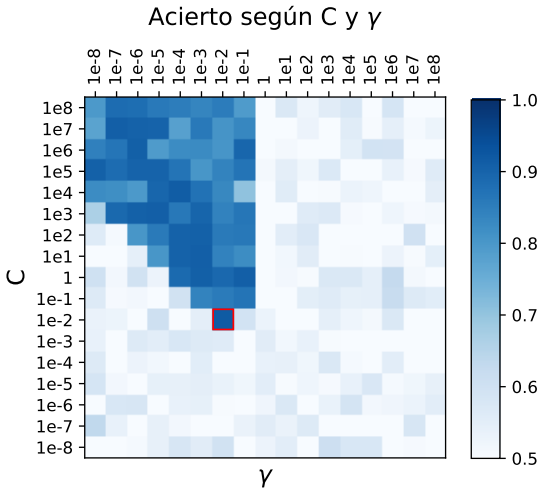

Figura 4.55: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo el valor de los hiperparámetros seleccionado por validación cruzada.

Aplicando la SVM con núcleo lineal se obtiene una tasa de acierto que mejoran en cerca de un 1.5 % las de los clasificadores entrenados con Random Forest. Los tiempos de ejecución son muy altos si los comparamos con los requeridos por el Random Forest. Esto se debe a que en el caso de RF no hemos realizado una validación cruzada para seleccionar los valores de los hiperparámetros.

Por último entrenamos una SVM con núcleo RBF. En este caso tenemos que seleccionar la mejor pareja de hiperparámetros C y γ. Los escogemos por validación cruzada. La pareja obtenida ha sido C$= \gamma = 10^{-2}$. Para poder ver cómo varían los aciertos en función de los valores de los hiperparámetros mostramos la Figura 4.55. Con esta configuración de los hiperparámetros obtenemos unas tasas de acierto del 80.9 % y 81.5 % en validación cruzada y test respectivamente. Se ha empleado un total de 2219.65 segundos en hacer la validación cruzada.

Puede verse cómo a partir de ciertos valores del hiperparámetro C la tasa acierto promedio se estanca. Lo contrario ocurre con γ. En este caso el acierto crece hasta que a partir de un cierto valor se desploma. Aplicando el test de Wilcoxon a los resultados con los de la SVM con núcleo lineal obtenemos que no podemos concluir que se aprecien diferencias significativas. Se tiene un $p-$valor de 0.2534, por lo que no podemos rechazar la hipótesis nula. A pesar de obtener este $p-$valor sí que se aprecia un claro aumento en el tiempo de ejecución con respecto a la SVM con núcleo lineal. Esto se debe a que el núcleo RBF requiere seleccionar dos hiperparámetros en lugar de uno.

Pasamos ahora a analizar el segundo método. Para generar el segundo conjunto de datos hemos empleado 10 componentes principales. Tomamos esta cantidad del número de componentes principales porque la tasa de acierto se mantiene constante para valores superiores, cómo puede verse en la Figura 4.56. Podríamos tomar más componentes principales para generar el conjunto de datos transformado pero no aumentaría la tasa de acierto.

Una vez tenemos los conjuntos entrenamos los Random Forest. Para hacernos una idea de cómo varían los aciertos en función de los valores del número de árboles del RF (τ) podemos ver la Figura 4.57. Como puede verse tomar como del hiperparámetro `n_estimators` 1023 sigue siendo razonable. Además se puede apreciar cómo la tasa de acierto tiende a un valor asintótico y deja de crecer. Con estas configuraciones de los hiperparámetros obtenemos una tasa de acierto en test del 79 %. Se ha requerido un tiempo de ejecución de 1.03 segundos en entrenar el clasificador.

Tras entrenar los RF con las componentes principales del conjunto original obtenemos una leve mejoría con respecto a la metodología anterior (de cerca del 0.7 %). Además los tiempos de ejecución se han reducido muy sutilmente. Esto se debe a que la dimensión de los datos se ha reducido.

Pasamos ahora a entrenar una SVM con núcleo lineal. Seleccionamos el valor del hiperparámetro C por validación cruzada. El valor del hiperparámetro escogido es C $= 10^{-4}$. Las tasas de acierto en validación cruzada y test son del 81.3 % y 81.7 %. Para realizar la validación cruzada se han necesitado 40.62 segundos. En la Figura 4.58 podemos ver cómo varía la tasa de acierto en función

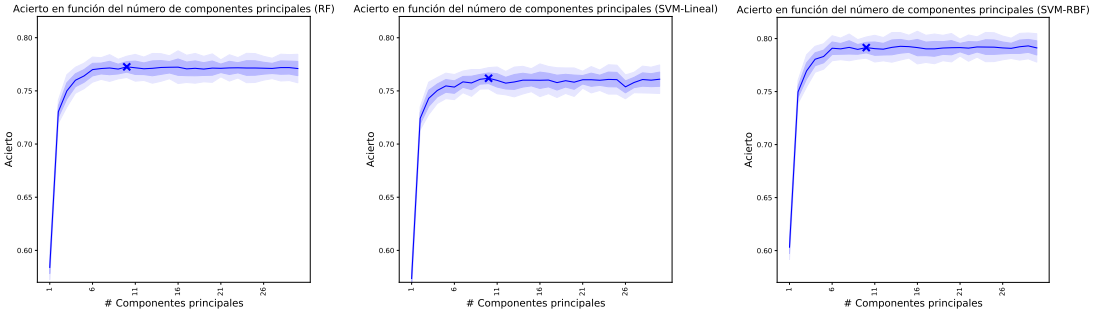

Figura 4.56: Tasa de acierto promedio ± una y dos desviaciones típicas de 50 Random Forest con $\tau = 1023$ (izquierda), SVM con núcleo lineal (centro) y SVM con núcleo RBF (derecha) en función del número de componentes principales empleadas para generar el conjunto transformado. Se han entrenado las SVM con los valores de los hiperparámetros obtenidos por validación cruzada (para cada número de componentes principales). Se marcan con una cruz la cantidad de componentes principales escogidas (10).

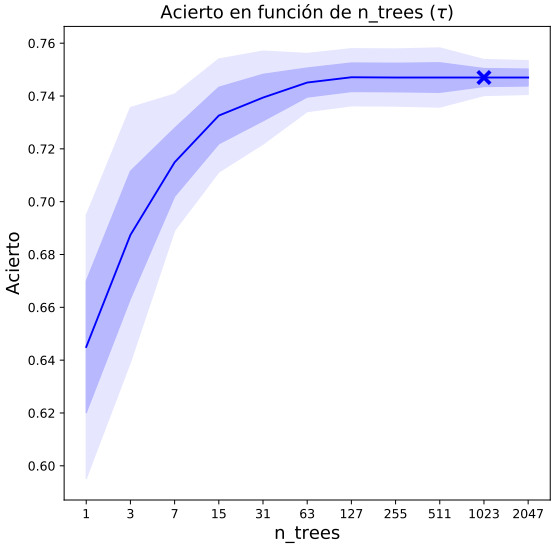

Figura 4.57: Tasa de acierto promedio en función de τ ± una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 12$. Se marca con una cruz el valor del hiperparámetro escogido.

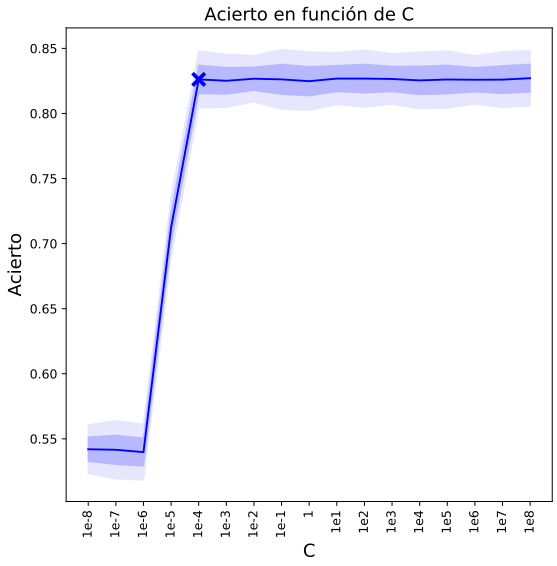

Figura 4.58: Tasa de acierto promedio en función de C ± una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

del valor del hiperparámetro C. El acierto se mantiene muy bajo para valores pequeños de C y crece rápidamente a partir de un valor umbral. Después sigue manteniéndose constante.

Con este clasificador se tienen unos resultados mejores que al utilizar la metodología anterior. Además esta diferencia es lo suficientemente grande cómo para haya una clara evidencia estadística que lo soporte. Aplicando el test de Wilcoxon obtenemos un $p-$valor igual a $62 \cdot 10^{-3}$. Los tiempos de ejecución empleados para seleccionar las configuraciones de los hiperparámetros se han reducido notablemente. Esto se debe a que se ha disminuido el número de atributos del conjunto.

Por último entrenamos una SVM con núcleo RBF. Seleccionamos los valores de los hiperparámetros C y γ por validación cruzada. Tenemos que la pareja de hiperparámetros escogida es $C = \gamma = 10^{-2}$. Las tasas de acierto en validación cruzada y test son del 82.2 % y 82.6 % en 889.72 segundos. En la Figura 4.59 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Para valores grandes de γ el error es muy alto. Lo mismo ocurre para valores pequeños de C. Sin embargo, cuando C es grande y γ pequeño la tasa de acierto es alta. Las mejores tasas de acierto se obtienen en un valor intermedio. Este valor coincide con las configuraciones óptimas de los hiperparámetros halladas por validación cruzada.

Se obtienen mejores tasas de acierto que al entrenar los mismos clasificadores con el conjunto original. Aplicando el test de Wilcoxon obtenemos evidencia estadística suficiente como para afirmar que existe una diferencia significativa a un nivel de significación del 90 %. El $p-$valor del contraste vale $53 \cdot 10^{-3}$. Además se puede apreciar cómo los tiempos de ejecución se han reducido en gran medida. Esto se debe a que hemos reducido notablemente el número de atributos.

A continuación utilizamos el tercer conjunto de datos. Para ello hemos proyectado las funciones originales en la base de Fourier con diferentes cantidades de elementos. Para entrenar los Random Forest usaremos 12 elementos en la base de Fourier, para las SVM con núcleo lineal 17 y para las SVM con núcleo RBF 18. Seleccionamos estas cantidades del parámetro n_F por validación cruzada. Cómo puede apreciarse en la Figura 4.60, proyectar las funciones en bases de Fourier con más elementos hace que disminuya la tasa de acierto (se añade ruido). Este acierto crece según aumenta el número de elementos en la base de Fourier hasta un cierto valor umbral. Después decrece lentamente según aumenta el parámetro n_F. En las Figuras 5.9 y 5.10 de la Sección 5.2.3 del apéndice pueden verse las gráficas de las elecciones por validación cruzada para los clasificadores SVM con núcleos lineal y RBF.

Para hacernos una idea de que aspecto tienen las funciones proyectadas en la base de Fourier, mostramos cinco funciones de cada clase. En la Figura 4.61 podemos ver cómo las funciones se han suavizado y parecen separarse ligeramente mejor en los instantes intermedios. Esto nos hace pensar que, con este conjunto de datos, se obtendrán mejores tasas de acierto que con el conjunto

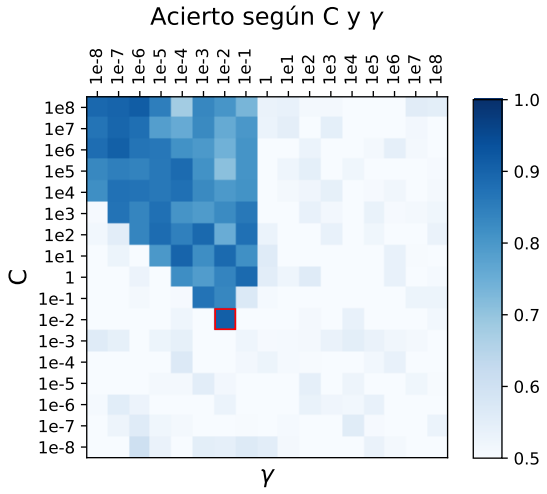

Figura 4.59: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo valores de los hiperparámetros escogidos por validación cruzada.

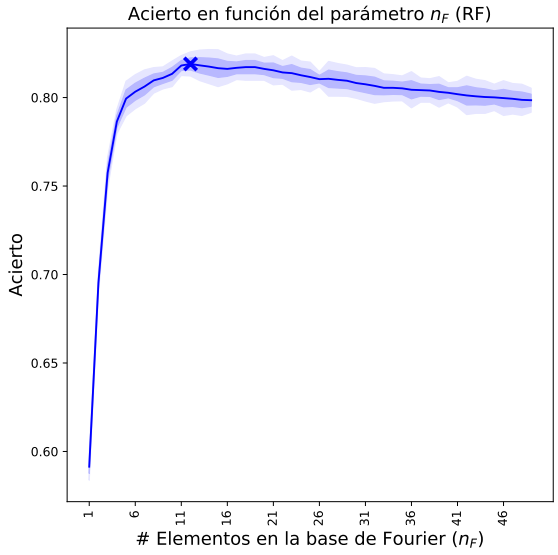

Figura 4.60: Tasa de acierto promedio $\pm$ una y dos desviaciones típicas de 50 Random Forest en función del número de elementos de la base de Fourier (n_F) empleados para generar el conjunto transformado. Se marca con una cruz el valor de n_F escogido por validación cruzada.

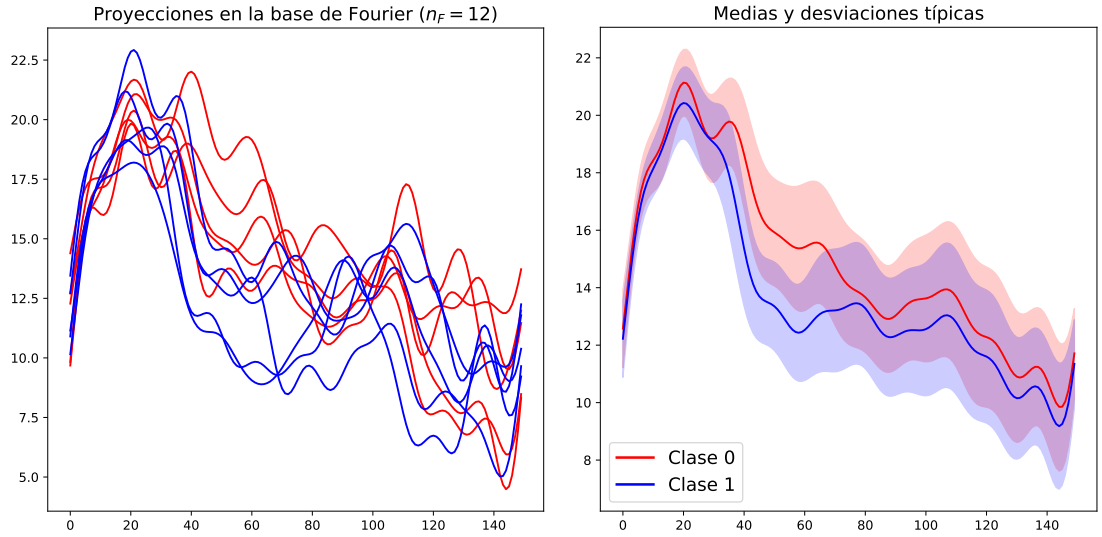

Figura 4.61: A la izquierda, las proyecciones de 5 funciones de cada clase del conjunto original en la base de Fourier con 12 elementos ($n_F = 12$). A la derecha las medias de cada clase $\pm$ una desviación típica de todo el conjunto proyectado en esta base.

original. Entrenamos los Random Forest. Para ello tomamos 1023 como valor del hiperparámetro `n_estimators`. En la Figura 4.62 podemos ver cómo esta elección sigue siendo razonable. Además aquí podemos observar cómo varían los aciertos en función de los valores de τ. La tasa de acierto crece según lo hace τ hasta estancarse en un valor asintótico. Obtenemos una tasa de acierto del 79.2 % en test en un total de 1.05 segundos. Podemos observar una leve mejoría con respecto a los aciertos conseguidos usando el conjunto de datos original y peores que cuando aplicamos la metodología de las componentes principales. Además, a pesar de trabajar con 25 atributos podemos representar las funciones proyectadas para hacernos una idea de su aspecto (véase la Figura 4.61). Esto es una clara ventaja con respecto a la metodología de las componentes principales (en este caso sólo podemos visualizar las transformaciones cuando se utilizan 2 o 3 componentes principales).

Ahora entrenamos una SVM con núcleo lineal. Escogemos el valor del hiperparámetro `C` por validación cruzada. El valor del hiperparámetro seleccionado es `C` $= 10^{-2}$. Las tasas de acierto en validación cruzada y test son del 82.3 % y 82.6 % respectivamente. Se han empleado 39.98 segundos en realizar la selección de los hiperparámetros por validación cruzada. En la Figura 4.63 podemos ver cómo varía la tasa de acierto en función de los diferentes valores del hiperparámetro `C` de la malla. La tasa de acierto es baja para valores pequeños de `C`. Llegado un cierto valor del hiperparámetro crece muy rápidamente para mantenerse constantes para valores superiores. Con las configuraciones de los hiperparámetros escogidos por validación cruzada obtenemos resultados superiores a los de los clasificadores entrenados usando la metodología de las componentes principales. Además se consigue evidencia estadística suficiente en el test de Wilcoxon que apoye esta afirmación con un nivel de significación del 95 %. El $p-$valor del contraste vale $4{,}2 \cdot 10^{-3}$.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros `C` y γ usamos una validación cruzada. La pareja de hiperparámetros escogida es `C` $= 10^{-1}$ y $\gamma = 10^{1}$. En la Figura 4.64 podemos ver cómo varían las tasas de acierto en función cada pareja de hiperparámetros `C` y γ. Los aciertos se mantienen altos para valores grandes de `C` y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. Las tasas de acierto de validación cruzada y test son del 83.1 % y 83.5 %. Se han empleado 944.33 segundos en realizar la selección de la pareja de hiperparámetros. Los resultados obtenidos son superiores a los que se tienen cuando utilizamos las SVM con núcleo lineal. Además las tasas de acierto son mayores que las que se tienen cuando entrenamos los clasificadores aplicando la metodología de las componentes principales. También se obtiene evidencia estadística suficiente para poder afirmar, aplicando el test de Wilcoxon, que hay una clara mejoría con respecto a la metodología original con un nivel de significación del 95 %. El $p-$valor del contraste vale $4{,}1 \cdot 10^{-3}$.

A continuación juntamos los dos conjuntos de datos anteriores (las componentes principales y

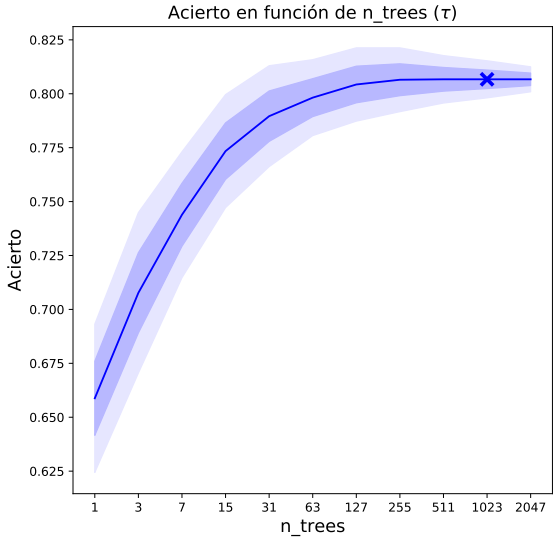

Figura 4.62: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 12$. Se marca con una cruz el valor del hiperparámetro seleccionado.

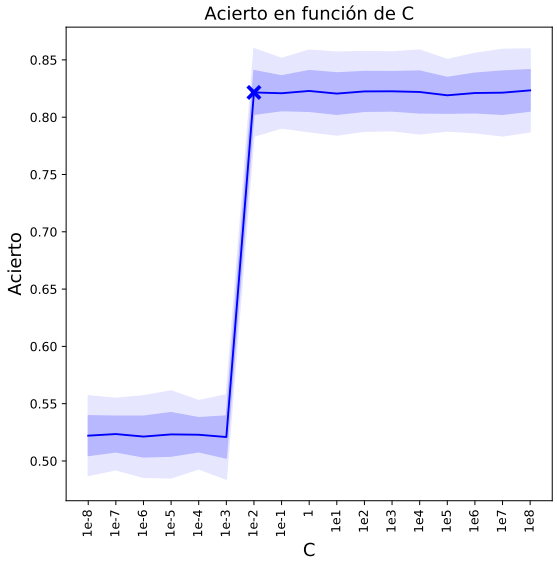

Figura 4.63: Tasa de acierto promedio en función de `C` $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogidos por validación cruzada.

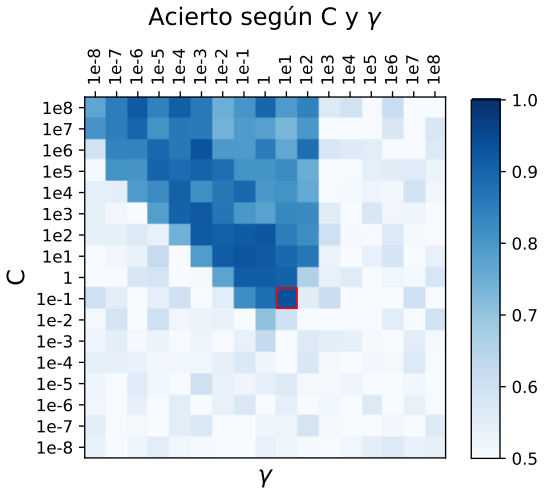

Figura 4.64: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

los coeficientes en las bases de Fourier). Con el conjunto agrupado entrenamos un Random Forest. Utilizamos el mismo valor del hiperparámetro τ. En la Figura 4.65 podemos observar cómo varían los aciertos en función de τ. También podemos ver cómo tomar $\tau = 1023$ sigue siendo una elección razonable. Puede apreciarse el mismo fenómeno que con los Random Forest anteriores. El acierto crece con τ hasta un valor asintótico. Obtenemos una tasa de acierto en test del 81.4 %. Se ha necesitado un total de 1.09 segundos para entrenar el clasificador.

Es con este conjunto de datos con el que se consigue la mejor tasas de acierto de entre todos los clasificadores Random Forest. La tasa de acierto aumentan en cerca de un 3 % con respecto a los resultados de la metodología original.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro C por validación cruzada. Tenemos que el valor del hiperparámetro escogido es $C = 10^{-3}$. En la Figura 4.66 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro C. Se tiene un error alto para valores pequeños de C. Este decrece rápidamente hasta mantenerse constante en valores de C superiores. Las tasa de acierto en validación cruzada y test es de 83.8 % y 84.2 % en 41.45 segundos respectivamente. Es aplicando esta metodología cuando obtenemos las mejores tasas de acierto de entre todas las SVM con núcleo lineal. Además obtenemos resultados significativamente mejores a un nivel de significación del 99 % si lo comparamos con los resultados de entrenar el clasificador con los conjuntos originales. Aplicando el test de Wilcoxon obtenemos que el $p-$valor del contraste vale $73 \cdot 10^{-4}$. Además el tiempo requerido para seleccionar las configuraciones de los hiperparámetros es bastante menor. Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros C y γ usamos una validación cruzada. Tenemos que la pareja de hiperparámetros escogida es $C = \gamma = 10^{-2}$. En la Figura 4.67 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Se puede apreciar el mismo comportamiento que en las SVM con núcleo RBF anteriores. Los aciertos son altos para valores grandes de C y pequeños de γ. en cualquier otro caso los aciertos son muy bajos. Hay un valor intermedio de los hiperparámetros donde el error es mínimo. Este valor intermedio coincide con las configuraciones obtenidas mediante validación cruzada. Las tasas de acierto en validación cruzada y test son del 84 % y 84.3 % en 1793.12 segundos. Usando esta metodología obtenemos las mejores tasas de acierto de todo el experimento. Los resultados son significativamente mejores que cuando se aplica el conjunto de datos original para entrenar los clasificadores para cualquier nivel de significación habitual. Aplicando el test de Wilcoxon obtenemos que el $p-$valor del contraste vale $86 \cdot 10^{-4}$. Por lo tanto hay evidencia estadística suficiente para afirmar que las tasas de acierto de los clasificadores entrenados usando la metodología de juntar las componentes principales y los coeficientes de Fourier son mejores que los obtenidos utilizando el conjunto original (a cualquier nivel de significación habitual).

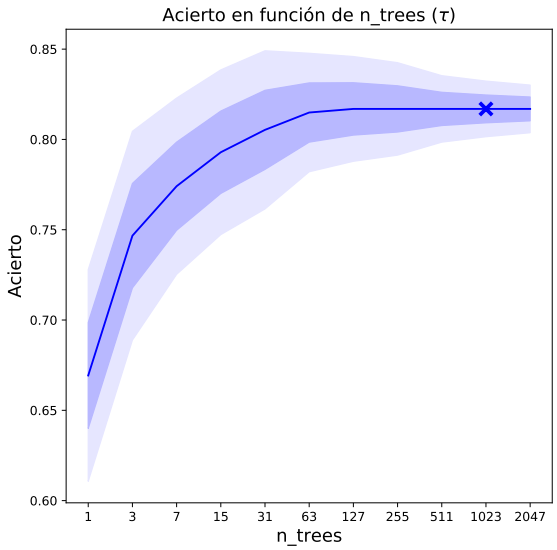

Figura 4.65: Tasa de acierto promedio en función de $\tau \pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 7$. Se marca con una cruz el valor del hiperparámetro seleccionado.

Pasamos a utilizar la técnica del troceado de las funciones siguiendo la metodología explicada en la Sección 3.3. Troceamos las funciones originales en varias subfunciones y entrenamos 50 Random Forest para seleccionar los mejores lugares por donde cortar las funciones originales. En la Figura 4.68 podemos ver las tasas de acierto promedio para cada lugar de corte de las funciones originales. Puede verse cómo, según se toman intervalos más pequeños el acierto disminuye para los instantes finales. El acierto es mayor para valores grandes de n y m entre 30 y 70 (esta es la zona donde mejor se separan las medias de las dos clases). La mejor tasa de acierto se tiene cuando se trocean las funciones originales por los instantes $[33, \ldots, 40]$. Por lo tanto los valores de n y m que utilizamos para generar las subfunciones del conjunto de datos troceado son $n = 33$ y $m = 40$. Una vez hemos troceado las funciones entrenamos un Random Forest. Utilizamos los mismos valores del hiperparámetro τ que en los casos anteriores. En la Figura 4.69 podemos observar cómo varía la tasa de acierto en función de los valores de τ de la malla. Además se puede apreciar cómo la elección de $\tau = 1023$ sigue siendo razonable. La tasa de acierto crece según lo hace τ hasta estancarse en un valor asintótico. Se obtiene una tasa de acierto del 75.1 % en test. Se ha empleado un total de 1.04 segundos. Con esta metodología de trocear las funciones originales obtenemos los peores resultados de todo el experimento. Estos resultados son significativamente peores que los obtenidos con la metodología original para cualquier nivel de significación. Obtenemos un $p-$valor usando el test de Wilcoxon de 81^{-4}. No obstante el tiempo de ejecución empleado en entrenar los clasificadores se reduce levemente. Esto se debe a que se disminuye el número de atributos de los datos.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro `C` por validación cruzada. Para cada valor de a tenemos que el valor del hiperparámetro escogido es $\texttt{C} = 10^{-4}$. En la Figura 4.70 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro `C`. La tasa de acierto es baja para valores pequeños de `C`. Llegado un cierto valor del hiperparámetro crece muy rápidamente para mantenerse constantes para valores superiores. La tasa de acierto de validación cruzada y test es del 78.1 % y 78.4 % en 40.08 segundos. Los aciertos de las SVM entrenados a partir de los datos originales troceados también obtienen las peores tasas de acierto de entre todas las metodologías utilizadas. Si los comparamos con la tasa de acierto del clasificador entrenado a partir del conjunto original, obtenemos evidencia estadística suficiente (en el test de Wilcoxon) para afirmar que, a un nivel de significación del 95 %, se obtienen peores resultados. El $p-$valor en este caso vale $23 \cdot 10^{-3}$.

Entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros `C` y γ usamos una validación cruzada. Para cada valor de a tenemos que la pareja de hiperparámetros escogida es $\texttt{C} = 10^{-2}$ y $\gamma = 10^{-1}$. En la Figura 4.71 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros `C` y γ. Los aciertos se mantienen altos para valores grandes de `C` y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiper-

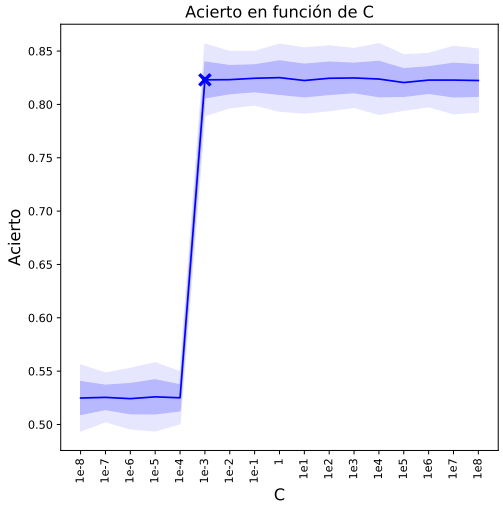

Figura 4.66: Tasa de acierto promedio en función de C ± una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

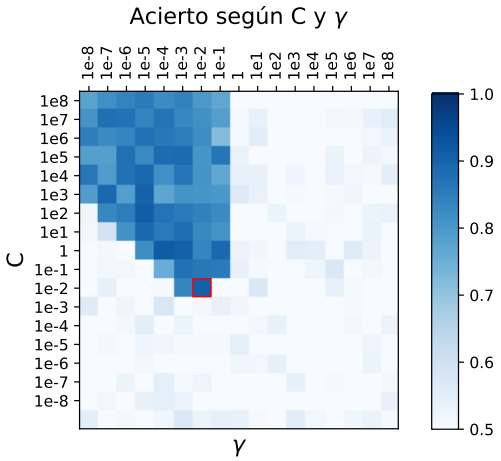

Figura 4.67: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

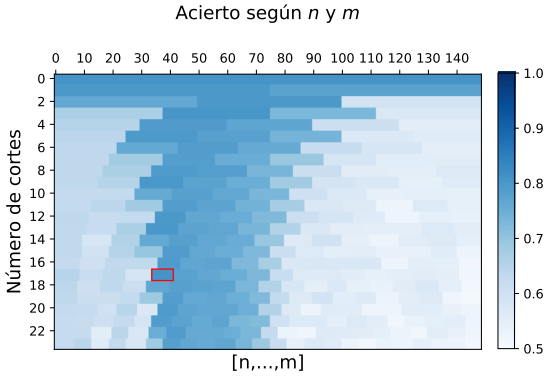

Figura 4.68: Tasas de acierto promedio de 50 Random Forest para los conjuntos troceados por los instantes $[n, \ldots, m]$.

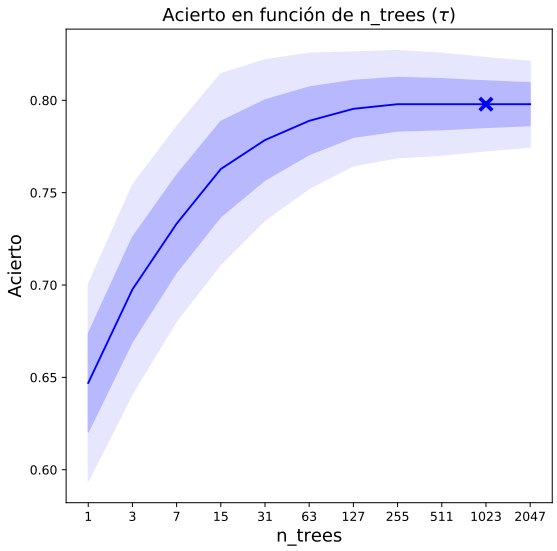

Figura 4.69: Tasa de acierto promedio en función de τ $\pm$ una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 12$. Se marca con una cruz el valor del hiperparámetro seleccionado.

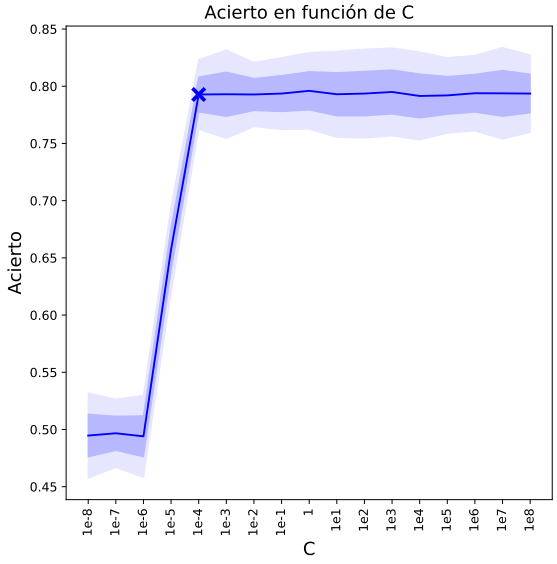

Figura 4.70: Tasa de acierto promedio en función de `C` $\pm$ una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

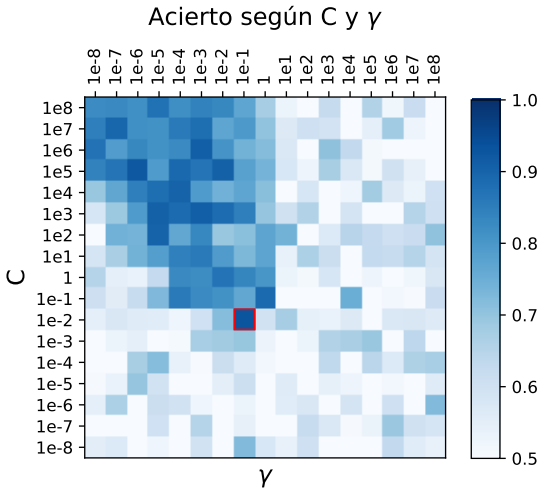

Figura 4.71: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF. Se marca con un cuadrado rojo el valor de los hiperparámetros escogido por validación cruzada.

parámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. La tasa de acierto promedio de validación cruzada y test es del 79.6 % y 80 % en 1004.50 segundos. Las tasas de acierto de estos clasificadores son peores que los que se obtienen al utilizar el conjunto de datos original. Aplicando el test de Wilcoxon podemos afirmar a un nivel de significación del 90 % que los resultados son peores que los que se obtienen aplicando la metodología original. En este caso el $p-$valor del contraste vale $94 \cdot 10^{-3}$.

Por último reducimos la dimensión de los datos mediante una selección de variables de los instantes de las funciones. Aplicamos la metodología explicada en la Sección 3.3. De esta manera obtenemos que vamos a seleccionar 11 variables (en las Figuras 5.11 y 5.12 de la Sección 5.2.3 del apéndice pueden verse las gráficas de las elecciones por validación cruzada para los clasificadores SVM con núcleos lineal y RBF). En la Figura 4.72 podemos ver las tasas de acierto de validación cruzada. Entrenamos un Random Forest. Utilizamos los mismos valores del hiperparámetro τ que en los casos anteriores. En la Figura 4.73 podemos observar cómo varía la tasa de acierto en función de los valores de τ de la malla. Además se puede apreciar cómo las elecciones de $\tau = 1023$ siguen siendo adecuadas. La tasa de acierto crece según lo hace τ hasta estancarse en un valor asintótico. De esta forma se obtiene una tasa de acierto en test del 78.8 %. Se ha requerido un total de 1.05 segundos. Utilizando esta metodología para entrenar los Random Forest obtenemos una tasa de acierto ligeramente superior que cuando usamos todo el conjunto original. Aunque no obtenemos evidencia estadística suficiente para concluir que los resultados son mejores a un nivel de significación elevado, sí podemos observar que los tiempos de ejecución han disminuido.

Ahora entrenamos una SVM con núcleo lineal. Tomamos el valor del hiperparámetro C por validación cruzada. Tenemos que el valor del hiperparámetro escogido es $\mathtt{C} = 10^{-2}$. En la Figura 4.74 podemos observar cómo varía la tasa de acierto en función del valor del hiperparámetro C. Las tasas de acierto son bajas para valores pequeños de C. Llegado un cierto valor del hiperparámetro crecen muy rápidamente para mantenerse constantes para valores superiores. Las tasas de acierto de validación cruzada y test es del 80.9 % y 81.2 % en 40.21 segundos . Los resultados que se obtienen al utilizar las SVM con núcleo lineal son similares a los que se consiguen con el conjunto original. Las tasas de acierto son ligeramente superiores y los tiempos de ejecución empleados en hallar las mejores configuraciones de los hiperparámetros se han reducido.

Por último entrenamos ahora una SVM con núcleo RBF. Para seleccionar los valores de los hiperparámetros C y γ usamos una validación cruzada. Tenemos que la pareja de hiperparámetros escogida es $\mathtt{C} = \gamma = 10^{-2}$. En la Figura 4.75 podemos ver cómo varía la tasa de acierto en función cada pareja de hiperparámetros C y γ. Los aciertos se mantienen altos para valores grandes de C y pequeños de γ. Sin embargo hay un valor intermedio de ambos hiperparámetros en el que al acierto es máximo. Estos valores coinciden con los escogidos por validación cruzada. Las tasas de acierto de validación cruzada y test son del 80 % y 80.5 % en 930.90 segundos. En este caso obtenemos resultados ligeramente superiores que cuando utilizamos la metodología original para entrenar los

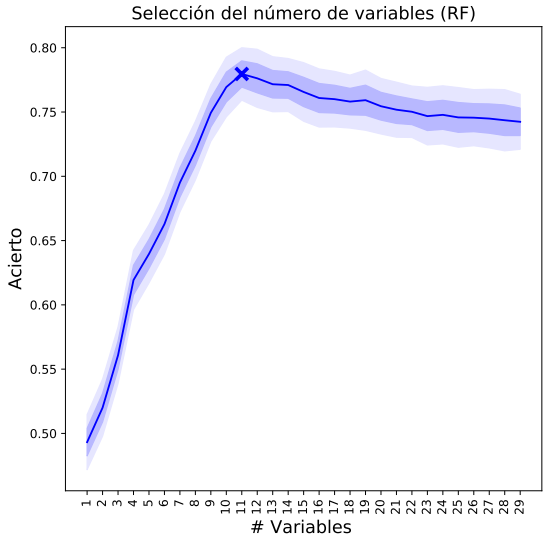

Figura 4.72: Tasas de acierto promedio en test y validación cruzada ± una y dos desviaciones típicas en función del número de variables seleccionadas. Se marcan con una cruz el número de variables seleccionadas por validación cruzada.

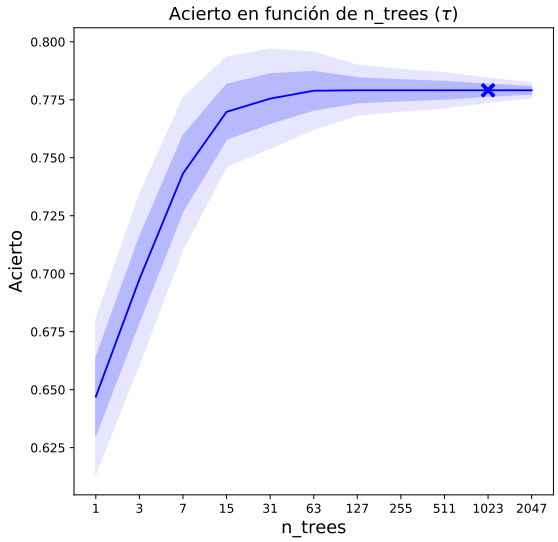

Figura 4.73: Tasa de acierto promedio en función de τ ± una y dos desviaciones típicas en 50 realizaciones de Random Forest con $\nu = 7$. Se marca con una cruz el valor del hiperparámetro seleccionado.

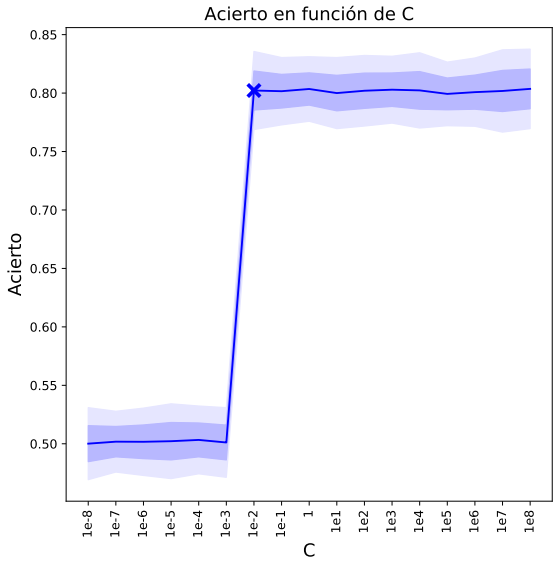

Figura 4.74: Tasa de acierto promedio en función de C ± una y dos desviaciones típicas en 50 SVM con núcleo lineal. Se marca con una cruz el valor del hiperparámetro escogido por validación cruzada.

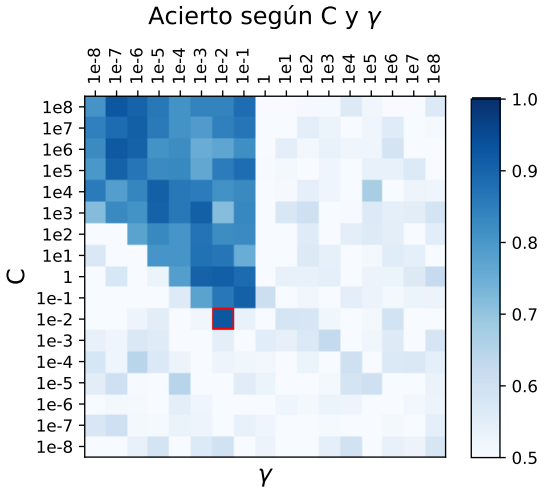

Figura 4.75: Tasa de acierto promedio en función de C y γ en 50 SVM con núcleo RBF para cada a. Se marcan con un cuadrado rojo los valores de los hiperparámetros escogidos por validación cruzada.

clasificadores. No se obtiene evidencia estadística suficiente para concluir esta afirmación a un nivel de significación elevado pero sí se puede apreciar una mejoría con respecto a las SVM con núcleo lineal.

A modo de resumen, mostramos a continuación una tabla con los resultados del experimento.

Algoritmo	Método	$\mu_{CV}(\sigma_{CV})$	$\mu_{test}(\sigma_{test})$	t
RF	Original	X	0.783 (0.008)	1.21
RF	10 componentes principales	X	0.790 (0.010)	1.03
RF	Fourier ($n_F = 12$)	X	0.792 (0.010)	1.05
RF	Agrupados	X	**0.814 (0.012)**	1.09
RF	Troceados	X	0.751 (0.008)	1.04
RF	11 coordednadas	X	0.788 (0.011)	1.05
SVM-Lineal	Original	0.801 (0.009)	0.806 (0.008)	46.31
SVM-Lineal	10 componentes principales	0.813 (0.013)	0.817 (0,013)*	40.62
SVM-Lineal	Fourier ($n_F = 17$)	0.823 (0.009)	0.826 (0,010)**	39.98
SVM-Lineal	Agrupados	0.838 (0.010)	**0.842** (0,009)***	41.45
SVM-Lineal	Troceados	0.781 (0.011)	0.784 (0.010)	40.08
SVM-Lineal	11 coordenadas	0.809 (0.010)	0.812 (0.009)	40.21
SVM-RBF	Original	0.809 (0.099)	0.815 (0.008)	2219.65
SVM-RBF	10 componentes principales	0.822 (0.010)	0.826 (0,010)*	889.72
SVM-RBF	Fourier ($n_F = 18$)	0.831 (0.010)	0.835 (0,009)**	944.33
SVM-RBF	Agrupados	0.840 (0.009)	**0.843** (0,008)***	1793.12
SVM-RBF	Troceados	0.796 (0.012)	0.800 (0.011)	1004.50
SVM-RBF	11 coordenadas	0.820 (0.009)	0.825 (0.009)	930.90

Cuadro 4.5: Tasas de acierto promedio y tiempos de ejecución de cada algoritmo en función de la metodología utilizada para el experimento `Phoneme`. Se marcan un *, ** y *** los resultados que son significativamente mejores en el test de Wilcoxon comparados con la metodología original a un nivel de significación del 90 %, 95 % y 99 % respectivamente. En negrita se marca el mejor resultado para cada familia de algoritmos.

Después de analizar este conjunto de datos obtenemos conclusiones similares a las del experimento anterior. Las tasas de acierto que se obtienen aplicando el algoritmo RF son mayores cuando se representan los datos funcionales en la base de Fourier, con respecto al RF entrenado con los datos originales y con las 10 componentes principales. Además, se puede observar que el tiempo de ejecución empleado por los RF con los datos transformados a la base de Fourier se ha con respecto al entrenado con los datos originales. Esto se debe a que, al representar los datos funcionales en esta base, además estamos realizando una reducción de la dimensión (pasando de trabajar con 152 variables a 25). Cuando aplicamos las SVM con núcleo lineal obtenemos clasificadores con tasas de acierto de entorno al 82 % en ambos enfoques. Sin embargo los tiempos de ejecución son diferentes. Gracias a representar los datos en la base de Fourier hemos conseguido reducir el tiempo de ejecución del algoritmo. Lo mismo ocurre con las SVM con núcleo RBF. Otra conclusión a la que llegamos es que las máquinas de vector soporte obtienen mejores tasas de acierto en test que los Random Forest. Al igual que ocurría en los dos experimentos anteriores, la metodología con la que se obtiene la mejor tasa de acierto en test consiste en agrupar las componentes principales y los coeficientes de Fourier de las funciones originales. Por último, a pesar de que la selección de variables y la metodología del troceado de las funciones no consigan buenas tasas de acierto, sí que son útiles para reducir la dimensión de los datos y para hacernos una idea de su comportamiento.

4.4. Conclusiones

Tras haber procesado y analizado los conjuntos de datos y haber estudiado el fundamento teórico que sustenta el problema de la clasificación supervisada de de datos funcionales hemos obtenido las siguientes conclusiones:

A la hora de resolver un problema de clasificación conviene tener en cuenta con qué tipo de datos vamos a trabajar. Podemos plantear el problema desde un punto de vista multivariante o con un enfoque funcional, para así poder explotar algunas de sus propiedades, como la continuidad, derivabilidad, posibles simetrías o el hecho de que podemos proyectar las trayectorias en una base de funciones como la de Fourier. Desde un punto de vista teórico ambas puntos de vista son muy diferentes, pero podría haber ocurrido que en la práctica no fuera así. A la vista de los resultados de los experimentos, puede verse cómo sí que hay diferencias en la práctica. La metodología de procesamiento con la que se han obtenido mejores resultados ha sido la del agrupamiento de las componentes principales y de los coeficientes en la base de Fourier, usando una combinación de los dos enfoques.

En general las SVM consiguen mejores resultados en términos de acierto en test que Random Forest. Además parece apreciarse una leve mejoría al pasar de utilizar un núcleo lineal a uno RBF. Sin embargo, los tiempos de ejecución se ven incrementados notablemente.

Otra conclusión a la que llegamos es que la metodología de la proyección de las funciones en la base de Fourier parece ser bastante útil. Esta metodología no solo sirve para reducir el posible ruido presente, sino que además ha demostrado ser una técnica de reducción de dimensión bastante interesante. En trabajos posteriores podría estudiarse más a fondo el comportamiento de las proyecciones en diferentes bases de los datos funcionales, como las bases de polinomios de Legendre, Laguerre o Hermite.

Capítulo 5

Apéndices

5.1. Demostración del Teorema 2.4

Para cada punto $X = x$, el error que comete el clasificador g en ese punto se escribe como

$$\begin{aligned}\mathbb{P}(g(X) \neq & Y \mid X = x) = 1 - \mathbb{P}(g(X) = y \mid X = x) = \\ &= 1 - (\mathbb{P}(g(X) = 1, Y = 1 \mid X = x) + \mathbb{P}(g(X) = 0, Y = 0 \mid X = x)) = \\ &= 1 - (\mathbb{1}_{\{g(X)=1\}}\mathbb{P}(Y = 1 \mid X = x) + \mathbb{1}_{\{g(X)=0\}}\mathbb{P}(Y = 0 \mid X = x)) = \\ &= 1 - (\mathbb{1}_{\{g(X)=1\}}\eta(x) + \mathbb{1}_{\{g(X)=0\}}(1 - \eta(x))).\end{aligned}$$

$$\boxed{\mathbb{P}(g(X) \neq Y \mid X = x) = 1 - (\mathbb{1}_{\{g(X)=1\}}\eta(x) + \mathbb{1}_{\{g(X)=0\}}(1 - \eta(x)))}$$

Si a esto le restamos el error asociado al clasificador de Bayes obtenemos:

$$\begin{aligned}\mathbb{P}(g(X) \neq & y \mid X = x) - \mathbb{P}(g^*(X) \neq Y \mid X = x) = \\ &= 1 - (\mathbb{1}_{\{g(X)=1\}}\eta(x) + \mathbb{1}_{\{g(X)=0\}}(1 - \eta(x))) - 1 + \\ &+ (\mathbb{1}_{\{g^*(X)=1\}}\eta(x) + \mathbb{1}_{\{g^*(X)=0\}}(1 - \eta(x))) = \\ &= \eta(x)(\mathbb{1}_{\{g^*(X)=1\}} - \mathbb{1}_{\{g(X)=1\}}) + (1 - \eta(x))(\mathbb{1}_{\{g^*(X)=0\}} - \mathbb{1}_{\{g(X)=0\}}).\end{aligned}$$

Si además tenemos en cuenta que

$$\mathbb{1}_{\{g(X)=0\}} = 1 - \mathbb{1}_{\{g(X)=1\}},$$

podemos observar cómo

$$\begin{aligned}\eta(x)(\mathbb{1}_{\{g^*(X)=1\}} - \mathbb{1}_{\{g(X)=1\}}) + (1 - \eta(x))(1 - \mathbb{1}_{\{g^*(X)=1\}} - 1 + \mathbb{1}_{\{g(X)=1\}}) = \\ = (2\eta(x) - 1)(\mathbb{1}_{\{g^*(X)=1\}} - \mathbb{1}_{\{g(X)=1\}}).\end{aligned}$$

$$\boxed{\mathbb{P}(g(X) \neq Y \mid X = x) - \mathbb{P}(g^*X) \neq Y \mid X = x) = (2\eta(x) - 1)(\mathbb{1}_{\{g^*(X)=1\}} - \mathbb{1}_{\{g(X)=1\}})}$$

Para ver que $\mathbb{P}(g(X) \neq Y \mid X = x) - \mathbb{P}(g^*(X) \neq Y \mid X = x)$ es mayor o igual a cero para todo $x \in \Omega$ distinguimos en dos casos:

1. Si en x ocurre que $\eta(x) > \frac{1}{2}$ entonces $g^*(x) = 1$ y se obtiene que:

$$\underbrace{\underbrace{(2\eta(x) - 1)}_{\geq 0}\underbrace{(\underbrace{\mathbb{1}_{\{g^*(X)=1\}}}_{=1} - \underbrace{\mathbb{1}_{\{g(X)=1\}}}_{=0 \text{ ó } 1})}_{\geq 0}}_{\geq 0}.$$

2. Si en x ocurre que $\eta(x) \leq \frac{1}{2}$ entonces $g^*(x) = 0$ y por lo tanto:

$$\underbrace{\underbrace{(2\eta(x) - 1)}_{< 0}\underbrace{(\underbrace{\mathbb{1}_{\{g^*(X)=1\}}}_{=0} - \underbrace{\mathbb{1}_{\{g(X)=1\}}}_{=0 \text{ ó } 1})}_{\leq 0}}_{\geq 0}.$$

Y cómo se cumple en ambos casos, se obtiene

$$\boxed{\mathbb{P}(g(X) \neq Y \mid X = x) - \mathbb{P}(g^*(X) \neq Y \mid X = x) = |2\eta(x) - 1|\mathbb{1}_{\{g^*(X) \neq g(X)\}} \geq 0}$$

Por último, si integramos esto con respecto a la medida de probabilidad $\mu(dx)$ se concluye que

$$\boxed{L(g) - L(g^*) = \int_{\Omega} |2\eta(x) - 1|\mathbb{1}_{\{g^*(X) \neq g(X)\}} \mu(dx) \geq 0}$$

■

5.2. Gráficas adicionales

5.2.1. Experimento Brownianos con distinta esperanza

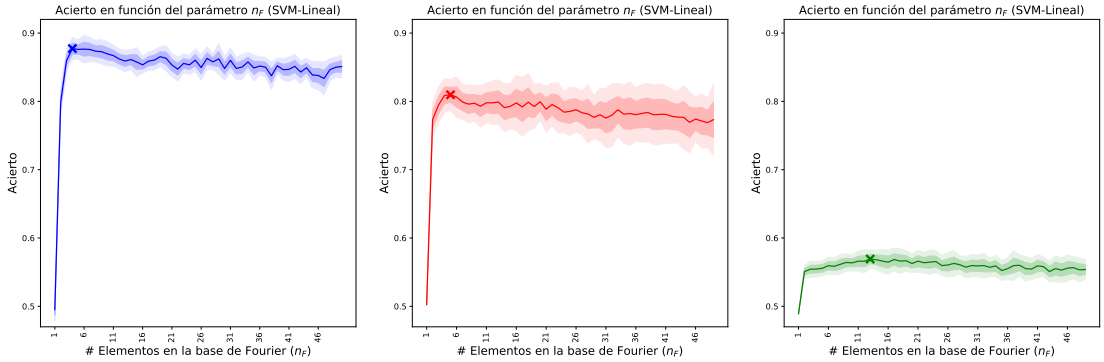

Figura 5.1: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de elementos de la base de Fourier (n_F) y para los tres valores de a. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo lineal.

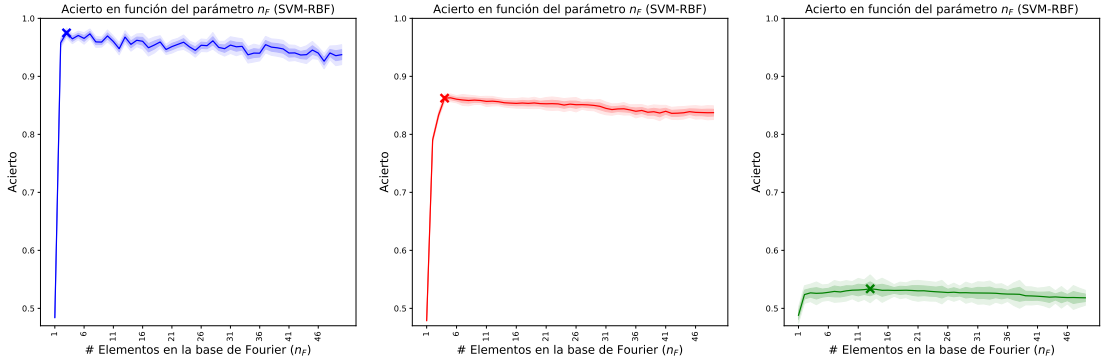

Figura 5.2: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de elementos de la base de Fourier (n_F) y para los tres valores de a. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo RBF.

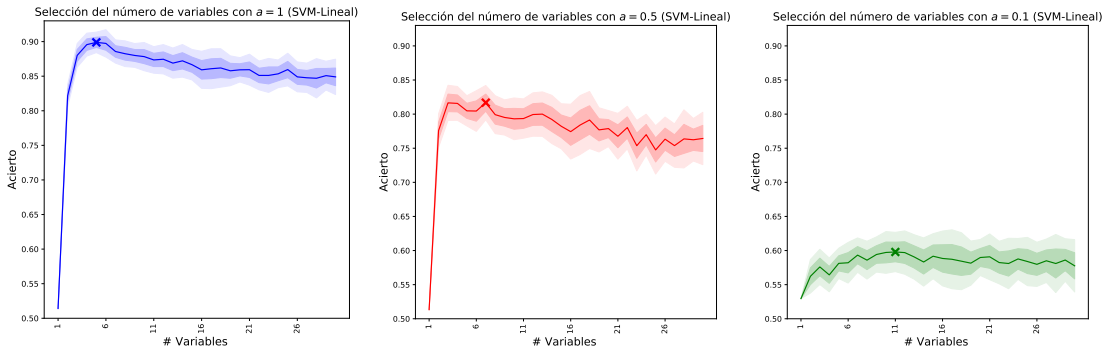

Figura 5.3: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de variables seleccionadas y para los tres valores de a. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo lineal.

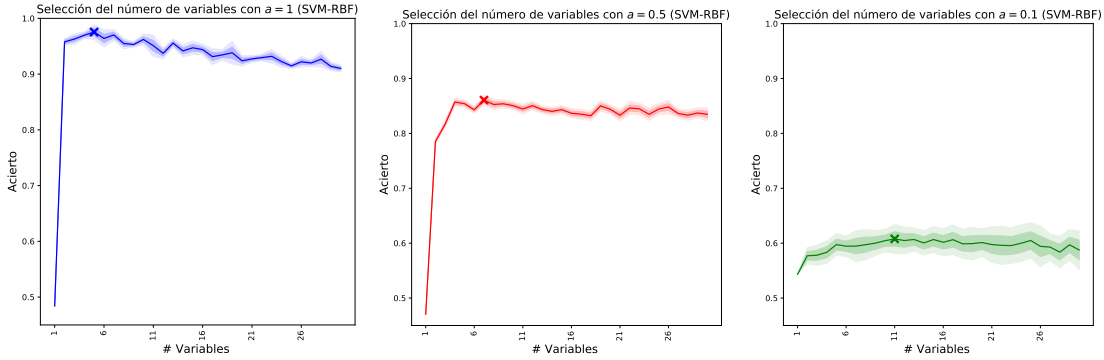

Figura 5.4: Tasa de acierto promedio en test y en validación cruzada $\pm$ una y dos desviaciones típicas para cada valor del número de variables seleccionadas y para los tres valores de a. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo RBF.

5.2.2. Experimento Berkeley

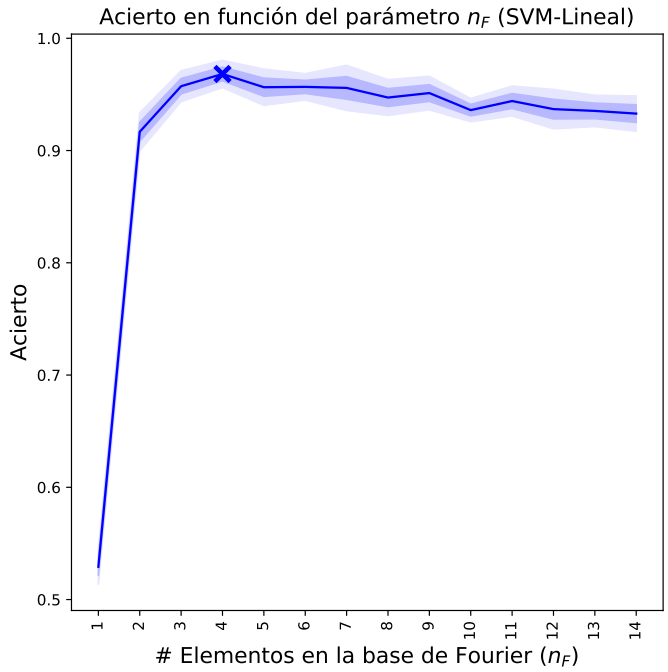

Figura 5.5: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de elementos de la base de Fourier (n_F. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo lineal.

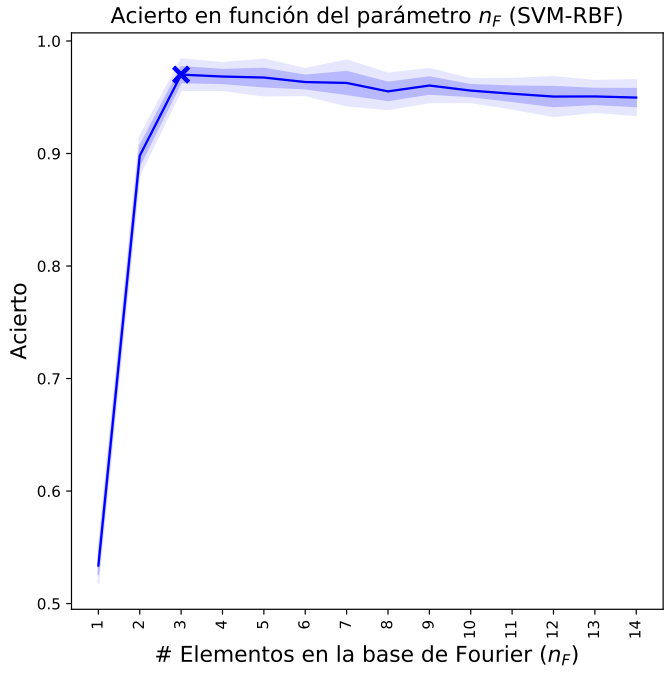

Figura 5.6: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de elementos de la base de Fourier (n_F). Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo RBF.

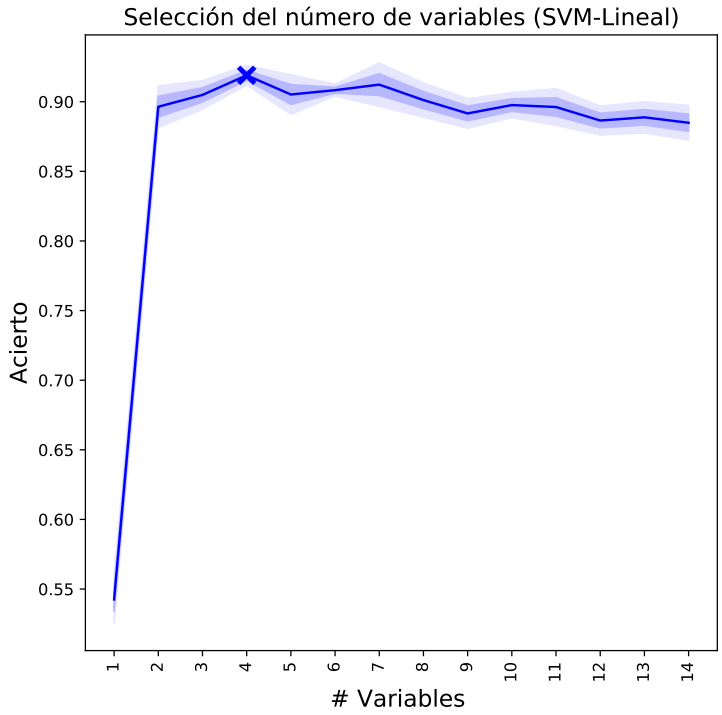

Figura 5.7: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de variables seleccionadas. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo lineal.

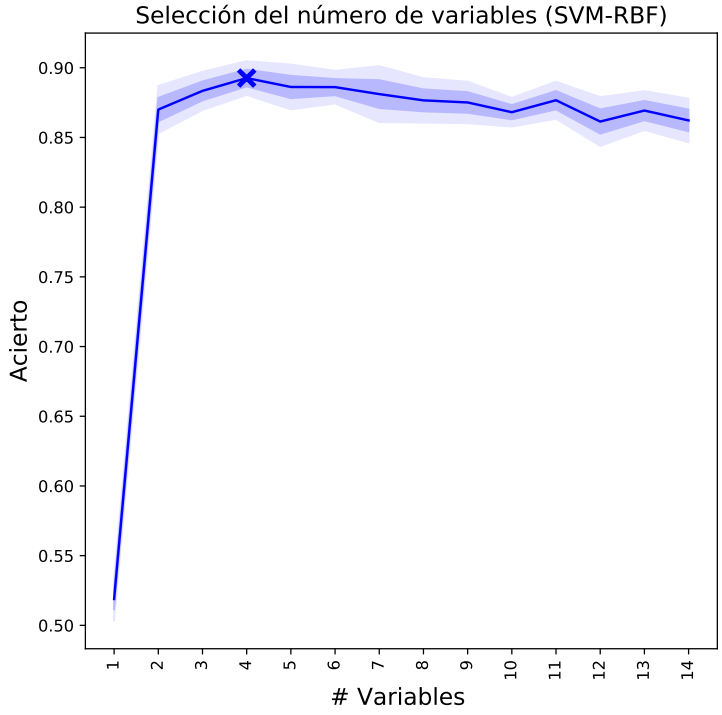

Figura 5.8: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de variables seleccionadas. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo RBF.

5.2.3. Experimento Phoneme

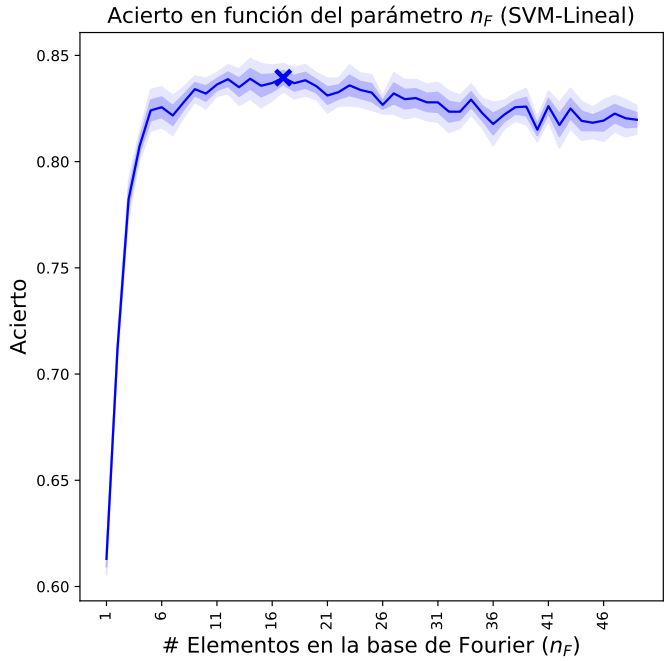

Figura 5.9: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de elementos de la base de Fourier (n_F. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo lineal.

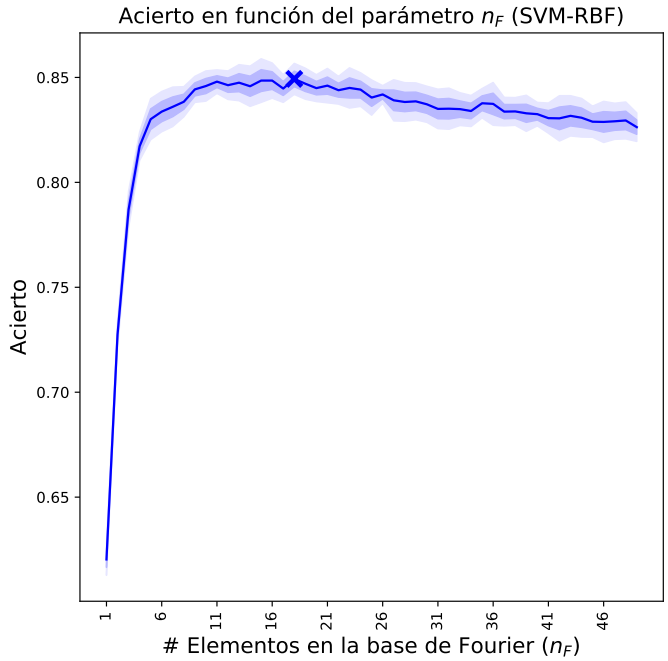

Figura 5.10: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de elementos de la base de Fourier (n_F). Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo RBF.

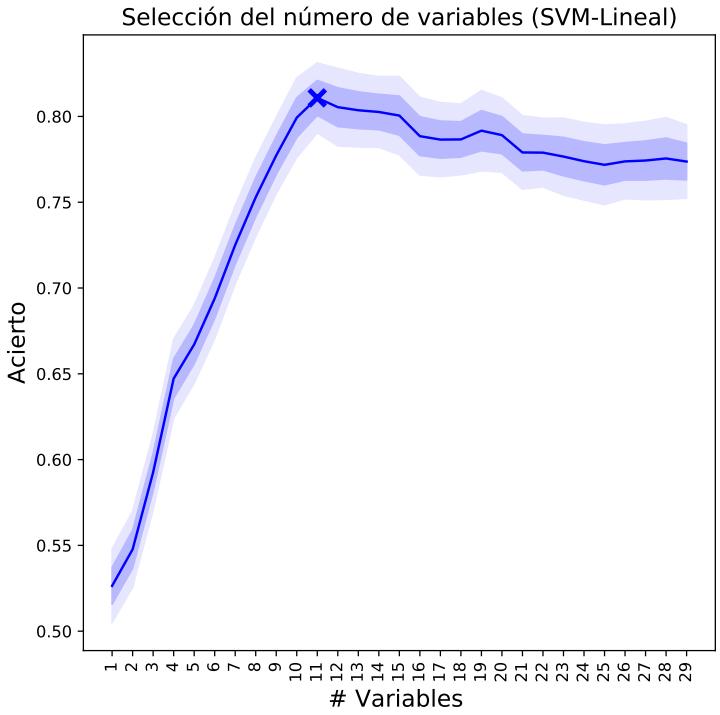

Figura 5.11: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de variables seleccionadas. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo lineal.

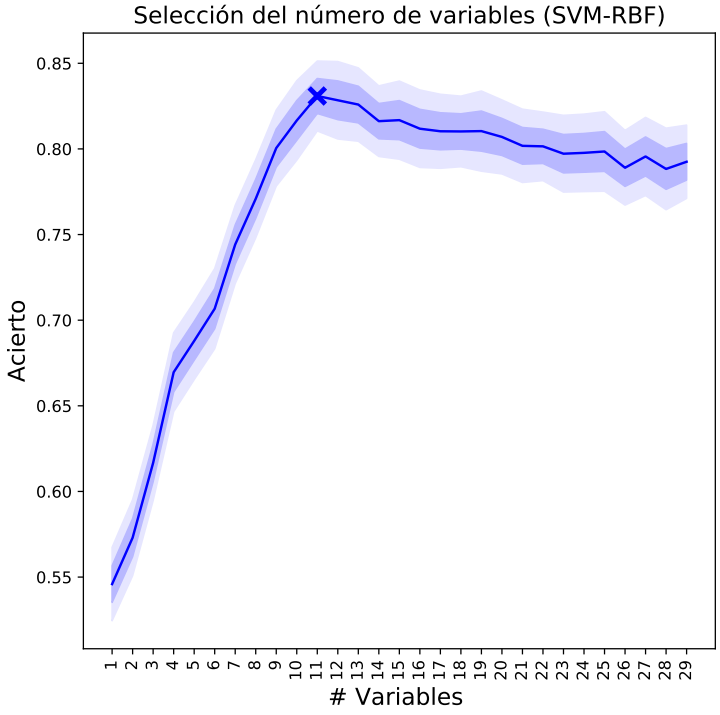

Figura 5.12: Tasa de acierto promedio en test y en validación cruzada ± una y dos desviaciones típicas para cada valor del número de variables seleccionadas. Se marcan con una cruz los valores de n_F escogidos por validación cruzada usando las SVM con núcleo RBF.

Bibliografía

[1] M. S. Alam and S. T. Vuong, *Random forest classification for detecting android malware*, in 2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing, IEEE, 2013, pp. 663–669.

[2] F. Beichelt, *Applied Probability and Stochastic Processes*, CRC Press, 2 ed., 2018.

[3] J. R. Berrendero, A. Cuevas, and J. L. Torrecilla, *On the use of reproducing kernel hilbert spaces in functional classification*, Journal of the American Statistical Association, 113 (2018), pp. 1210–1218.

[4] G. Biau, *Analysis of a random forests model*, Journal of Machine Learning Research, 13 (2012), pp. 1063–1095.

[5] G. Biau, L. Devroye, and G. Lugosi, *Consistency of random forests and other averaging classifiers*, Journal of Machine Learning Research, 9 (2008), pp. 2015–2033.

[6] A. Bosch, A. Zisserman, and X. Munoz, *Image classification using random forests and ferns*, in 2007 IEEE 11th international conference on computer vision, Ieee, 2007, pp. 1–8.

[7] D. Bosq, *Linear processes in function spaces: theory and applications*, vol. 149, Springer Science & Business Media, 2012.

[8] L. Breiman, *Random forests*, Machine learning, 45 (2001), pp. 5–32.

[9] C. Cai, L. Han, Z. L. Ji, X. Chen, and Y. Z. Chen, *Svm-prot: web-based support vector machine software for functional classification of a protein from its primary sequence*, Nucleic acids research, 31 (2003), pp. 3692–3697.

[10] Y.-d. Cai and S. L. Lin, *Support vector machines for predicting rrna-, rna-, and dna-binding proteins from amino acid sequence*, Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics, 1648 (2003), pp. 127–133.

[11] L. S. Calvo, *Algoritmo de random forest aplicado a problemas de clasificación supervisada*, 2018. Trabajo de Fin de Grado.

[12] C. Cortes and V. Vapnik, *Support-vector networks*, Machine learning, 20 (1995), pp. 273–297.

[13] A. Cuevas, *A partial overview of the theory of statistics with functional data*, Journal of Statistical Planning and Inference, 147 (2014), pp. 1–23.

[14] A. Cuevas, *Advanced course in statistics*, 2020.

[15] A. Cuevas, M. Febrero, and R. Fraiman, *Robust estimation and classification for functional data via projection-based depth notions*, Computational Statistics, 22 (2007), pp. 481–496.

[16] F. S. de Diego, *Apuntes de la asignatura variable real*, 2018.

[17] A. Dembo and K. Ross, *Stochastic processes*, 2013.

[18] J. Demšar, *Statistical comparisons of classifiers over multiple data sets*, Journal of Machine learning research, 7 (2006), pp. 1–30.

[19] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer, 2 ed., 1996.

[20] G. Fan, J. Cao, and J. Wang, *Functional data classification for temporal gene expression data with kernel-induced random forests*, in 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, IEEE, 2010, pp. 1–5.

[21] F. Ferraty and Y. Romain, *The Oxford handbook of functional data analaysis*, Oxford University Press, 2011.

[22] F. Ferraty and P. Vieu, *Non-parametric Functional Data Analysis. Theory and Practice.*, Springer, 1 ed., 2006.

[23] T. W. Gamelin, *Complex Analysis*, Springer, 1 ed., 2000.

[24] V. Gómez-Verdejo, M. Verleysen, and J. Fleury, *Information-theoretic feature selection for functional data classification*, Neurocomputing, 72 (2009), pp. 3580–3589.

[25] C. Heil, *Functional analysis lecture notes. weak and weak* convergence.*

[26] T. K. Ho, *Random decision forests*, in Proceedings of 3rd international conference on document analysis and recognition, vol. 1, IEEE, 1995, pp. 278–282.

[27] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112, Springer, 2013.

[28] R. Rahman, S. R. Dhruba, S. Ghosh, and R. Pal, *Functional random forest with applications in dose-response predictions*, Scientific reports, 9 (2019), p. 1628.

[29] S. Raj and K. C. Ray, *A comparative study of multivariate approach with neural networks and support vector machines for arrhythmia classification*, in 2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE), IEEE, 2015, pp. 1–6.

[30] F. Rossi and N. Villa, *Support vector machine for functional data classification*, Neurocomputing, 69 (2006), pp. 730–742.

[31] W. Rudin, *Real and Complex Analysis*, Mc Graw-Hill, 3 ed., 1987.

[32] B. Schölkopf, A. J. Smola, F. Bach, et al., *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.

[33] T. Shi, D. Seligson, A. S. Belldegrun, A. Palotie, and S. Horvath, *Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma*, Modern Pathology, 18 (2005), p. 547.

[34] S. Wager, *Asymptotic theory for random forests*, arXiv preprint arXiv:1405.0352, (2014).

[35] A. Wang et al., *An industrial strength audio search algorithm.*, in Ismir, vol. 2003, Washington, DC, 2003, pp. 7–13.

[36] Z. Xue, P. Du, and H. Su, *Harmonic analysis for hyperspectral image classification integrated with pso optimized svm*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7 (2014), pp. 2131–2146.